DOCUMENT DTI-UM-2769-2

USER MANUAL

FOR

DATA TRANSLATION INCORPORATED

DT2769

REAL-TIME CLOCK

DATA TRANSLATION

100 Locke Drive, Marlboro, MA 01752/(617) 481-3700

TABLE OF CONTENTS

...../cont.

TABLE OF CONTENTS (Cont.)

CHAPTER 1

INTRODUCTION

1.1 General Description

The DT2769 interface board is a Real-Time Clock system for users of the Digital Equipment Corporation (DEC) LSI-11 series of microcomputers. The DT2769 Real-Time Clock system is constructed on a standard DEC dual-height card that plugs directly into the LSI-11, LSI-11/2, PDP-11/03, or Heath H11 backplane. The DT2769 Real-Time Clock is totally compatible with the DEC KWV-11 Real-Time Clock but occupies only half the board space.

The Data Translation Real-Time Clock boards offer the user a full 16 bit event/interval counter that can be programmed to count at 5 internal crystal controlled frequencies, an external input, or from the Bus Event Line (50/60 Hz). In addition the counter can be programmed to operate in any one of the following 4 modes:

1) Single Interval
2) Repeated Interval
3) External Event Timing
4) External Event Timing from Zero Base

Thus offering the user the greatest flexibility in programming the Real-Time Clock.

The Data Translation Real-Time Clock boards also include two Schmitt triggers each of which can be set to fire at any level between ±12V and on either the positive or negative slope. These Schmitt triggers allow the user to start the clock, initiate A/D conversions, or generate program interrupts in response to external events.

User interfacing for Schmitt triggers and clock overflow signals is accomplished through the multi-pin connector (J1). In addition to appearing on (J1) the clock overflow pulse and Schmitt trigger pulse also appear on tab connectors allowing the user to jumper them directly to other boards via module jumpers.

The selection of the Control-Status Register address, interrupt vector address, and Schmitt trigger slope and level conditions are done through DIP switches. In addition provision is also made through the I/O connector J1 for external user provided slope switches and level controls for both Schmitt triggers.

CHAPTER 2

SPECIFICATIONS

2.1 The following specifications for the Data Translation Real-Time Clock boards are @ 25°C unless otherwise specified.

2.1.1 Clock

Oscillator Accuracy 0.01%

Range	Crystal Oscillator Base Frequency = 10MHz Divided into 5 rates = (1MHz, 100KHz, 10KHz, 1KHz, 100Hz) Other sources = line frequency, Schmitt trigger 1 input

2.1.2 Input Output Signals

All inputs and outputs are TTL compatible unless otherwise specified.

2.1.2.1 Input Signals

1. STI IN (Schmitt Trigger 1 Input)

Maximum Input Range	±30V
Triggering Range	±12V (settable by means of potentiometer) or TTL
Triggering Slope	User selectable by means of switch for either positive or negative slope.
Origin	User device
Response Time	For analog level depends on waveform and amplitude for TTL logic levels typically 600nS
Hysteresis	Approximately 0.5V, positive and negative
Characteristics	Single ended (input 100 KΩ impedance to ground)

2. ST2 IN (Schmitt Trigger 2 Input)

Same as ST1 IN

2.1.2.2 Output Signals

1. CLK OVL (Clock Overflow)

Asserted Level	Low
Destination	User device (A/D system typically)
Duration	Approximately 500nS
Characteristics	TTL open collector driver with 470 Ω pull-up to +5V. Maximum source current from output through load to ground when output is high ($\geq$ 2.4V) is 5mA.

Characteristics (cont.) Maximum sink current from external source voltage through load to output when output is low ($\leq$ 0.8V) is 8mA

2. ST1 OUT (Schmitt Trigger 1 Output)
   Same as CLK OVL

3. ST2 OUT (Schmitt Trigger 2 Outputs)
   Same as CLK OVL

2.1.2.3 Power Requirements (From Bus Power Supply)

DT2769: +5V. 1.75 Amps. MAX.
+12V 10mA Typical

CHAPTER 3

PROGRAMMING REGISTERS

3.1 Introduction

The interface registers of the DT2769 are designed to meet the requirements of standard DEC PDP-11 interfaces. As such, they are structured around a Control and Status Register for complete software control of the interface. The registers of the DT2769 are bit for bit compatible with the DEC KWV-11 Real-Time Clock. This chapter describes in detail the bit definitions of the Real-Time Clock registers.

3.2 Summary of Registers

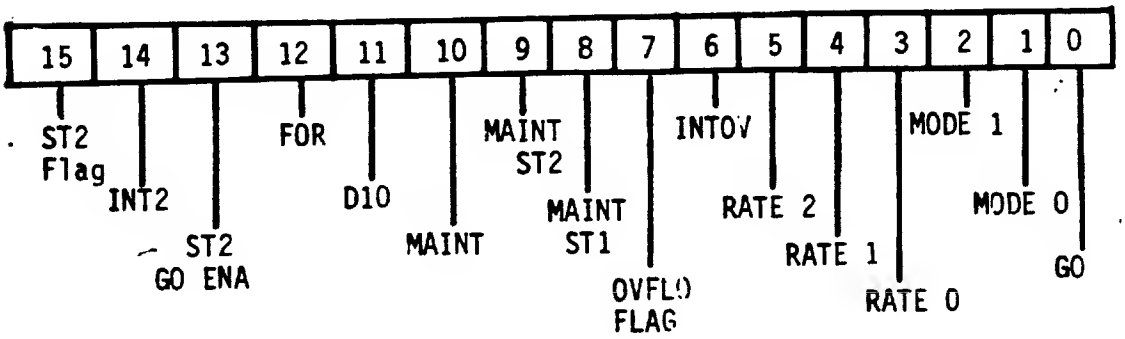
The following is a brief summary of the registers used by DT2769. The term base refers to the device address (See Chapter 4).

Register Name	Address	Comments
Control-Status Register (CSR)	Base	R/W Byte Addressable
Buffer/Preset Register (BPR)	Base + 2	R/W Word Addressable Only

3.3 Detailed Register Description

The following sections describe the interface registers in detail and are the same for both the DT2769 and DT1709.

3.3.1 Control-Status Register (CSR) Byte Addressable Base



Upper Byte

ST2 FLAG (bit 15) Read/Write to Zero

This bit is set by the firing of Schmitt Trigger 2 or by setting MAINT ST2 (bit 9) in any mode while GO (bit 0) or ST2 GO ENA (bit 13) is set. This bit is cleared under program control or when the GO bit is set unless the ST2 GO ENA bit had previously been set. The bit must be cleared after servicing an ST2 interrupt OVFLO FLAG (bit 7) interrupt if both are simultaneously in the enabled state. This bit is initialized to zero.

Chapter 3
Continued.

INT2 (bit 14) Read/Write

When set, this bit enables interrupts to be generated upon the ST2 FLAG (bit 15) becoming set. This bit can be set and cleared under program control and is cleared by bus init.

ST2 GO ENA (bit 13) Read/Write

When set will allow the assertion of ST2 FLAG (bit 15) to set the GO bit (bit 0) and simultaneously clear ST2 GO ENA (bit 13). This bit is set and cleared under program control and is also cleared when the GO bit (bit 0) is set or by bus init.

FOR (bit 12) Read/Write

This bit is set if the counter overflows while OVF FLAG (Bit 7) is set or if ST2 is fired while ST2 FLAG is set indicating for either case an overrun condition in which the hardware is running faster than the software can handle. This bit can be set and cleared under program control and is also cleared by setting the GO bit or by bus init.

D10 (bit 11) Read/Write

Setting this bit disables the internal crystal oscillator and is used in conjunction with bit 10 (MAINT. OSC) for maintenance and diagnostic purposes. This bit is set and cleared under program control.

MAINT OSC (bit 10) Write/Read as Zero

Setting this bit simulates one cycle of the internal 10MHz crystal oscillator. This bit is typically used by the diagnostic programs to test the clock counter and divider chain for proper operation. This bit is set under program control and always reads back as zero.

MAINT ST2 (bit 09) Write/Read as Zero

Setting this bit simulates the firing of ST2. All digital functions initiated by ST2 can thus be exercised under program control. This bit is also typically used by diagnostic programs. The bit is set under program control and alwyas reads back as zero.

MAINT ST1 (bit 08) Write/Read as Zero

Setting this bit simulates the firing of ST1. All digital functions initiated by ST1 firing can thus be exercised under program control. This bit is set under program control and always reads back as zero.

OVFLO FLAG (bit 07) Read/Write to Zero

This bit is set whenever the clock/counter register overflows indicating the occurrence of the overflow. This bit can be cleared under program control when the GO bit is set or by bus init.

INTOV (bit 06) Read/Write

When set this bit enables the assertion of OVFLO FLAG to generate an interrupt to the processor. This bit is set and cleared under program control.

CHAPTER 3

(Continued)

RATE(2-0) (bits 5-3) Read/Write
These bits are decoded to select the input time base for the clock/counter register from one of eight sources as outlined in the following table.

RATE 2 (bit 5)	RATE 1 (bit 4)	RATE 0 (bit 3)	RATE SELECTED
0	0	0	STOP
0	0	1	1MHz
0	1	0	100KHz
0	1	1	10KHz
1	0	0	1KHz
1	0	1	100Hz
1	1	0	ST1
1	1	1	BEVNT (50/60Hz LTC)

The RATE select bits are set and cleared under program control and cleared by bus init.

MODE (1-0) (bits 2-1) Read/Write
These bits are decoded to select one of four possible modes for clock operation as outlined in the following table:

MODE 1 (bit 2)	MODE 0 (bit 1)	MODE SELECTED
0	0	MODE 0 Single Interval
0	1	MODE 1 Repeated Interval
1	0	MODE 2 External Event Timing
1	1	MODE 3 External Event Timing From Zero Base

Chapter 3
Continued

These bits are set and cleared under program control and are also cleared by bus init.

GO (bit 0) Read/Write

Setting this bit initiates all counter on this board as determined by the setting of the RATE and MODE select bits. When cleared this bit causes the counter to be inhibited and cleared. The bit can be set and cleared under program control, in addition, it can also be set when ST2 FLAG is set while ST2 GO ENA is set. This bit stays set in MODE 1, 2 and 3 operation until explicitly cleared by the program. This bit can also be cleared when the counter overflows in MODE 0 operation or by bus init.

3.3.2 Buffer/Preset Register (BPR) Word Addressable Only

The BPR is a 16 bit, word addressable only, read/write register. This register is used in two different ways depending on the mode of operation selected. In MODE 0 and 1 operation, this register is loaded with the 2's complement of the number, of the number of clock counts desired before counter overflow. In MODE 2 and 3 operation this register provides indirect reading of the clock counter.

CHAPTER 4

OPERATION AND PROGRAMMING

4.1 Base Address Selection

The Base Address of the DT2769 which is the I/O address assigned to the Control-Status Register (CSR) is user selectable by means of DIP switches SW1 and SW2 located near the bus interface logic as shown on Figure 5.0. The Buffer/Preset Register address is then always two locations greater than the CSR (Base) address. The DIP switches allow the user to set the base address anywhere in the 170000_8 - 177774_8 address space in increments of 4_8. The recommended base address for the DT2769 is 170420_8 and is the address set at the factory. Figure 4.1 shows how switches SW1 and SW2 must be set to generate this base address.

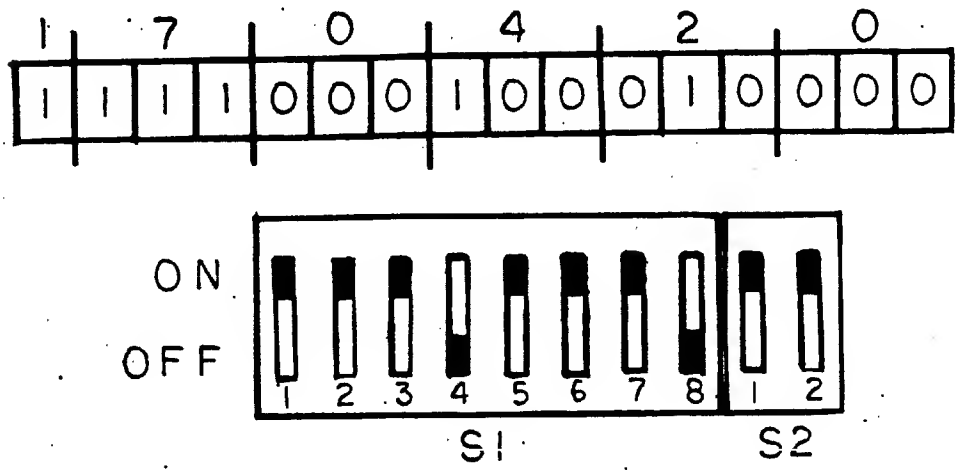


Figure 4.1

Switch Setting for Base Address of 170420_8

As shown in Figure 4.1 a switch in the "ON" position represents a "1" in the corresponding bit location and a switch in the "OFF" position represents a "0" in the corresponding bit location. Consequently switches a, 2, 3, 5, 6 and 7 on SW1 and switches 1 and 2 on SW2 are in the "OFF" position signifying zeroes while switches 4 and 8 on SW1 are in the "ON" position signifying ones.

CHAPTER 4
(Continued)

4.2 Vector Address Selection

The DT2769 Real-Time Clock system is capable of generating two distinct interrupt vectors to the LSI-11 processor. These interrupts can occur when,

1. Schmitt Trigger #2 is Fired
2. Clock/Counter Overflows

Each of these two events can generate a unique interrupt to the processor with the internal priority being arranged such that the clock overflow interrupt has the higher priority of the two if both occur simultaneously and are both enabled. The interrupt vector address of the clock overflow interrupt can be assigned any address in the 000_8 - 770_8 vector address space in 10_8 increments. The interrupt vector of the ST2 interrupt is then always 4_8 locations higher than the clock overflow interrupt address. The recommended interrupt vector address for the DT2769 is 440_8 and is set to that value at the factory. Figure 4.2 shows how switch SW2 must be set to generate this address.

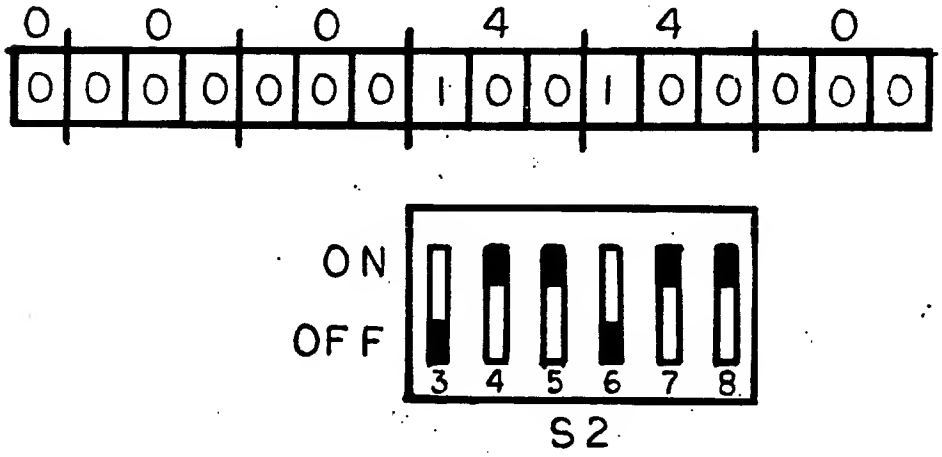


Figure 4.2
Switch Setting For Vector Address of 440_8

CHAPTER 4
(Continued)

4.3 Operation Modes

The DT2769 Real-Time Clock system can be programmed to run in four distinct modes making the Real-Time Clock useful for a variety of different applications such as interval generation, event counting, pulse generator, etc. The Clock Operating modes are:

1) Single Interval - Mode 0
2) Repeated Interval - Mode 1
3) External Event Timing - Mode 2
4) External Event Timing from Zero Base - Mode 3

In the following subsection the use of each of the four operating modes will be described in detail.

4.3.1 Single Interval - (Mode 0)

This mode of operation is used to generate a fixed interval for such applications as known delays. In this mode of operation the user loads the BPR with the 2' complement of the required number of clock pulses to be counted that will generate the amount of time delay required at the user selected clock frequency. For example, loading the BPR with -100 at a clock frequency of 1KHz would generate a 100ms time delay. The user then sets the GO bit either directly or by a Schmitt Trigger 2 event with bit 13 of the CSR set. When the GO bit becomes asserted the clock/counter is loaded with the value held in the BPR and then clocked at the selected frequency until overflow accurs. When the clock overflows the overflow bit (bit 7) of the CSR is set which can generate an interrupt to the processor if enabled or be checked under program control; clock overflow also causes the GO bit to be reset thus turning the clock off.

4.3.2 Repeated Interval (Mode 1)

This Mode of operation is similar to Mode 0 operation in that the BPR is loaded with the 2' complement of the desired clock counts required before overflow. The user then sets the GO bit and (either directly or via ST2) wait for the clock overflow. In this mode however the clock overflow will not clear the GO bit but instead reloads the counter from the BPR and continues counting without loss of any clock pulse. This process continues until the user clears the GO bit. In this way the user can generate a fixed frequency pulse train (overflow pulses) with any period within the range of the clock counter and the 5 crystal frequencies. For example loading the BPR with -1 and selecting 1MHz operation would generate a 1MHz pulse train. In general the overflow rate is equal to the oscillator rate divided by the absolute value of the present register.

4.3.3 External Event Timing (Mode 2)

This mode of operation is used to determine the time between a sequence of external events. In this Mode the user sets the GO bit which causes the clock/counter to be cleared and then incremented at the selected rate until the GO bit is cleared by the user under program control. The clock counter thus started will continue to free run (count) until explicitly turned off. Now each firing of ST2 by an external event will cause the instantaneous value of the clock/counter to be

CHAPTER 4

(Continued)

latched in the BPR and the ST2 FLAG to be set. The ST2 FLAG can be used to inform the program of the external event through interrupt or programmed testing of the ST2 FLAG. The program can then read the BPR and record the time of each occurrence of the event. During this time the clock is still free running after each event and after overflow, interrupts on overflow may be enabled to alert the program of each overflow condition.

4.3.4 External Event Timing From Zero Base (Mode 3)

This mode of operation is identical to Mode 2 operation except that the counter is reset to zero after the occurrence of each event.

4.4 Typical Program Sequence

In this section typical programming sequences are outlined for operating the Real-Time Clock in each of the four modes of operation.

4.4.1 Single Interval

1) User program loads BPR with 2's complement of desired clock count.

2) User program sets CSR for Mode 0 operation and selects appropriate rate.

3) User program sets the GO bit or sets ST2 GO ENA and waits for an external event to set the GO bit. When the GO bit gets set the counter is loaded and enabled.

4) Counter increments until overflow, then clears the GO bit and stops.

5) Clock overflow causes the Overflow Flag to be set which can issue an interrupt if INTOV (CSR bit 6) has been previously set, or waits for program intervention if interrupts had not been enabled.

6) The program responds to the interrupt or responds as a result of other events (i.e. testing the Overflow Flag or the A/D Done Flag if overflow was used to start and A/D conversion). The program reads the CSR, clears the Overflow Flag, and if no counting or mode changes are required, sets the GO bit or ST2 GO ENA bit to reenter the sequence at step 3.

4.4.2 Repeated Interval

1) User program loads BPR with the 2's complement of the word count.

2) Program sets the CSR for Mode 1 operation at selected frequency.

3) User program sets the GO bit or sets the ST2 GO ENA bit and waits for an external event to set the GO bit. When the GO bit gets set the counter is loaded and enabled.

4) The clock/counter increments until an overflow occurs.

5) When clock overflows the clock/counter is reloaded from the BPR and reenabled to count. The Overflow Flag is set which will generate an interrupt to the processor if enabled.

CHAPTER 4

(Continued)

6) If a second overflow occurs before first is serviced then the Flag Over Run (FOR) bit of the CSR is set alerting the processor of a loss of data.

7) Program responds to overflow by reading the CSR, clearing the Overflow Flag and if no changes are required sets the GO bit or the ST2 GO ENA bit to reenter the sequence of step 3.

4.4.3 External Event Timing (Mode 2)

1) Program selects Mode 2 operation the desired rate and sets the GO bit in the CSR.

2) The DT2769 or DT1709 responds by clearing the counter incrementing it at the selected rate until the GO bit is cleared.

3) Now ST2 pulses will latch the instantaneous reading of the clock/counter into the BPR and also set the ST2 FLAG which will generate an interrupt if enabled. Program can go and read current value of BPR to record time of event.

4) Clock continues to count even after overflow, however overflow will set Overflow Flag and generate an interrupt if enabled.

5) Process continue until program clears GO bit.

4.4.4 External Event Timing From Zero Base (Mode 3)

The operation here is identical to Mode 2 except that the clock/counter is cleared after every ST2 pulse.

4.5 USER INTERFACING

A 40-pin Berg connector (J1) has been provided for user inputs and and outputs. In addition to this connector two extra tab connectors have been provided that bring out the clock overflow and ST1 outputs. These tabs are electrically in parallel with pins RR (CLK OVL) and UU (ST1 OUTL) of the 40-pin connector (J1) and are intended to facilitate connections by means of module jumpers to the clock overflow and external start inputs on the DT2762 analog input board. The pin outs on connector (J1) are given in figure 4.0a. This connector is exactly compatible with the DEC KWV11-A

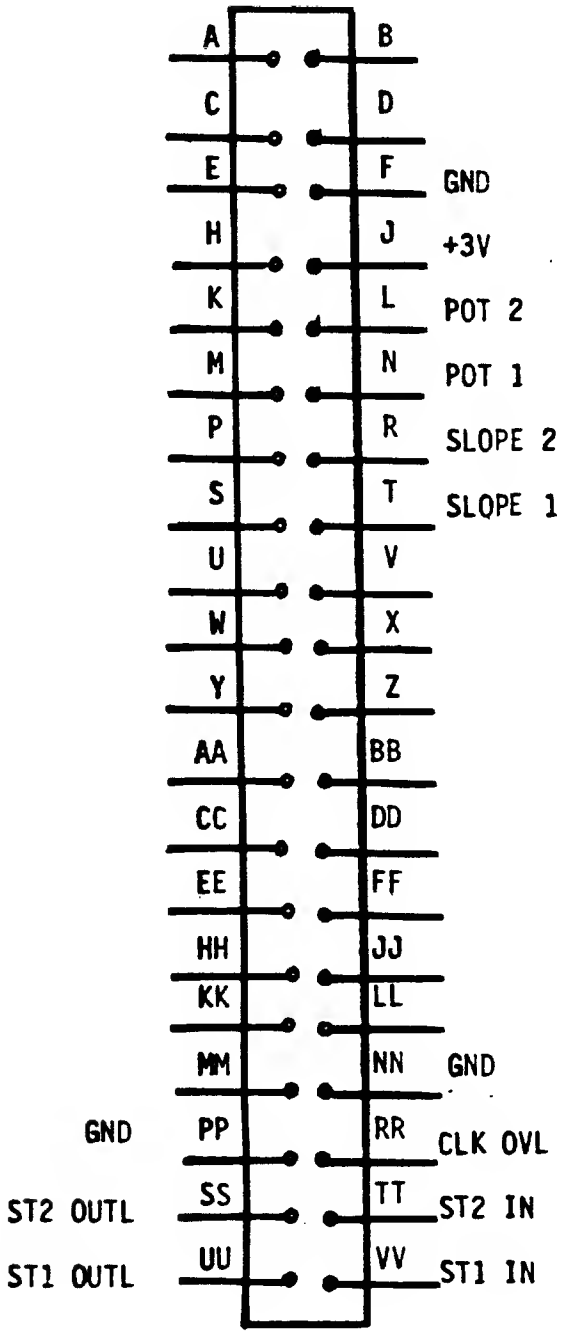
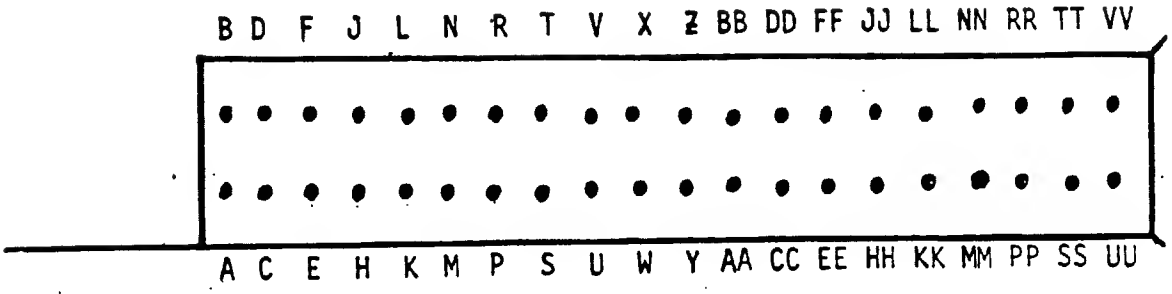


FIGURE 4.5a

The pin locations for connector J1 when looking into the connector are shown below:



CHAPTER 5

SCHMITT TRIGGER OPERATION

5.1 Introduction

The DT2769 Real-Time Clock system has on board two user adjustable Schmitt Triggers. These Schmitt Triggers are used to condition input waveforms to a form that can be used by the DT2769. Both Schmitt Triggers can be adjusted to trigger at any level in the ±12 volt range by means of the potentiometers labled ST1 LVL ADJ and ST2 LVL ADJ as shown on Figure 5.0. The Schmitt Triggers can also be set to trigger at TTL thresholds and on either the positive or negative slope of the input signal. The details on how the Schmitt Trigger firing level and slopes are set is given in Section 5.3.

The two Schmitt Triggers are identical in operation but perform totally different functions. The two Schmitt Triggers are labeled ST1 and ST2 and will be referred to by these names in the remainder of this manual.

5.2 Schmitt Trigger Usage

The two Schmitt Triggers provided on the DT2769 interface are operated and programmed in identical manners as described in the next section. The use of the two Schmitt Triggers are somewhat different as described below:

ST1 The output pulses of ST1 can be counted by the clock counter hence ST1 can be used to generate a time base for the clock counter if a periodic signal is attached to its input. ST1 can also be used to count events if ST1 is fired at each event and is selected as the clock input (Rate = 110). ST1 outputs also appear on the Faston tab labeled ST1 OUT as shown in Figure 5.0 and on pin uu of connector J1. This pulse may be used as a trigger to the DT2762 A/D system or any other user application.

ST2 The output pulses of ST2 are used to control clock operation in the following two ways:

1. Turns Clock On
2. Saves Counter Reading in BPR

Each of the two usages of ST2 are described below in detail.

1. Turning Clock On - When the ST2 GO ENABLE bit (CSR bit 13) is set, firing ST2 in any mode sets the GO bit and initiates counter action, causes the ST2 FLAG (CSR bit 15) to be asserted, and generates an interrupt if enable.
2. Saving Counter Reading - When the GO bit is set in Modes 2 and 3, firing ST2 causes the BPR to be loaded from the counter, the ST2 FLAG to be set, and an interrupt to be generated if enabled.

The ST2 pulse also appears on pin SS of connector J1 and may be used for other external applications.

Chapter 5
(Continued)

5.3 Schmitt Trigger Slope/Level Adjustments

The technique and methods for setting the slope and level for both Schmitt Triggers is identical. Hence the discussion will encompass both Schmitt Triggers simultaneously.

Slope and level adjustments for the Schmitt Triggers are done by use of Switch SW3 and the two potentiometers labelled ST1 LVL ADJ and ST2 LVL ADJ. Each Schmitt Trigger has associated with it three switches on SW3 and a potentiometer. The usages of these switches and the potentiometer are outlined in Table 5.3. Figure 5.1 shows how the slope terminology is used while Figure 5.2 shows how the switch SW3 is wired.

SW3 SWITCH NO.	USAGE
1	When this switch is on and switch 2 is off the firing level of ST1 is set by means of the ST1 LVL ADJ potentiometer anywhere in the ±12 volt range.
2	When this switch is on and switch 1 is off the firing level of ST1 is set to the TTL level and the potentiometer has no effect. Note: Both switches 1 and 2 should not be on together.
3	When this switch is on and switch 4 is off the firing level of ST2 is set by means of the ST2 LVL ADJ potentiometer anywhere in the ±12 volt range.
4.	When this switch is on and switch 3 is off the firing level of ST2 is set to the TTL level and the potentiometer has no effect. Note: Switches 3 and 4 should not be on together.
5.	When this switch is in the OFF position ST1 will fire on the negative slope of the input signal (high to low transition for a square wave). In the ON position ST1 will fire on the positive slope of the input signal (low to high transition of a square wave.)
6.	When this switch is in the OFF position ST2 will fire on the negative slope of the input signal. In the ON position ST2 will fire on the positive slope of the input signal.
7-8	Not used.

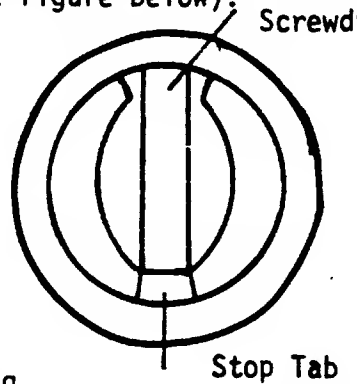
Table 5.3

5.4 External Control of Schmitt Trigger

The 40 pin user connector J1 provides connections that allow the user to set the Schmitt trigger slope and firing levels from external user supplied circuitry. In order to use the external circuitry the following board configuration is required.

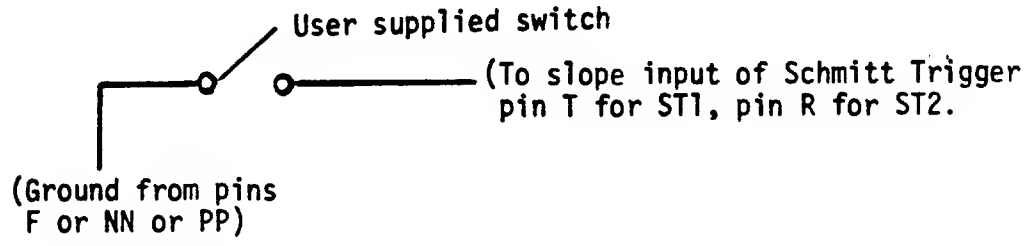
I If slope is to be externally controlled then the slope switch on board must be set to the off position. If external slope control is not required then the user should set this switch to the desired slope setting.

II If the triggering level is to be externally controlled then the variable slope switch for the Schmitt trigger must be selected. Note that the TTL firing level is not externally controllable. The user should also make sure that the onboard level potentiometers are close to the center of their ranges. This is accomplished by lining up the screwdriver slot of the potentiometer with the stop tab (see figure below).



External Slope Setting

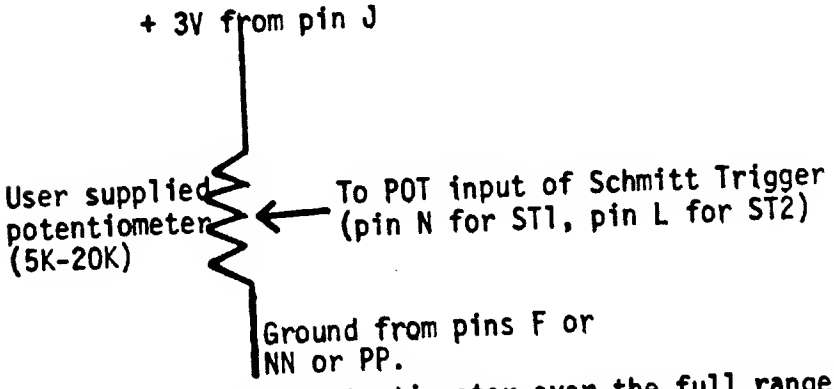
To control the slope of a Schmitt trigger the user requires a single switch which is connected as shown below:



External Level Control

External level control over the entire ±12 volt range requires a user supplied potentiometer. The value of the potentiometer should not be less than 5K and should not be greater than 20K.

Any potentiometer in this range will be adequate. The number of turns of the potentiometer is not critical, however, a potentiometer with greater turns will allow for a finer level adjust. The connection of the external potentiometer is shown below:



Then varying the potentiometer over the full range will change the firing level over the full ±12 volt range.

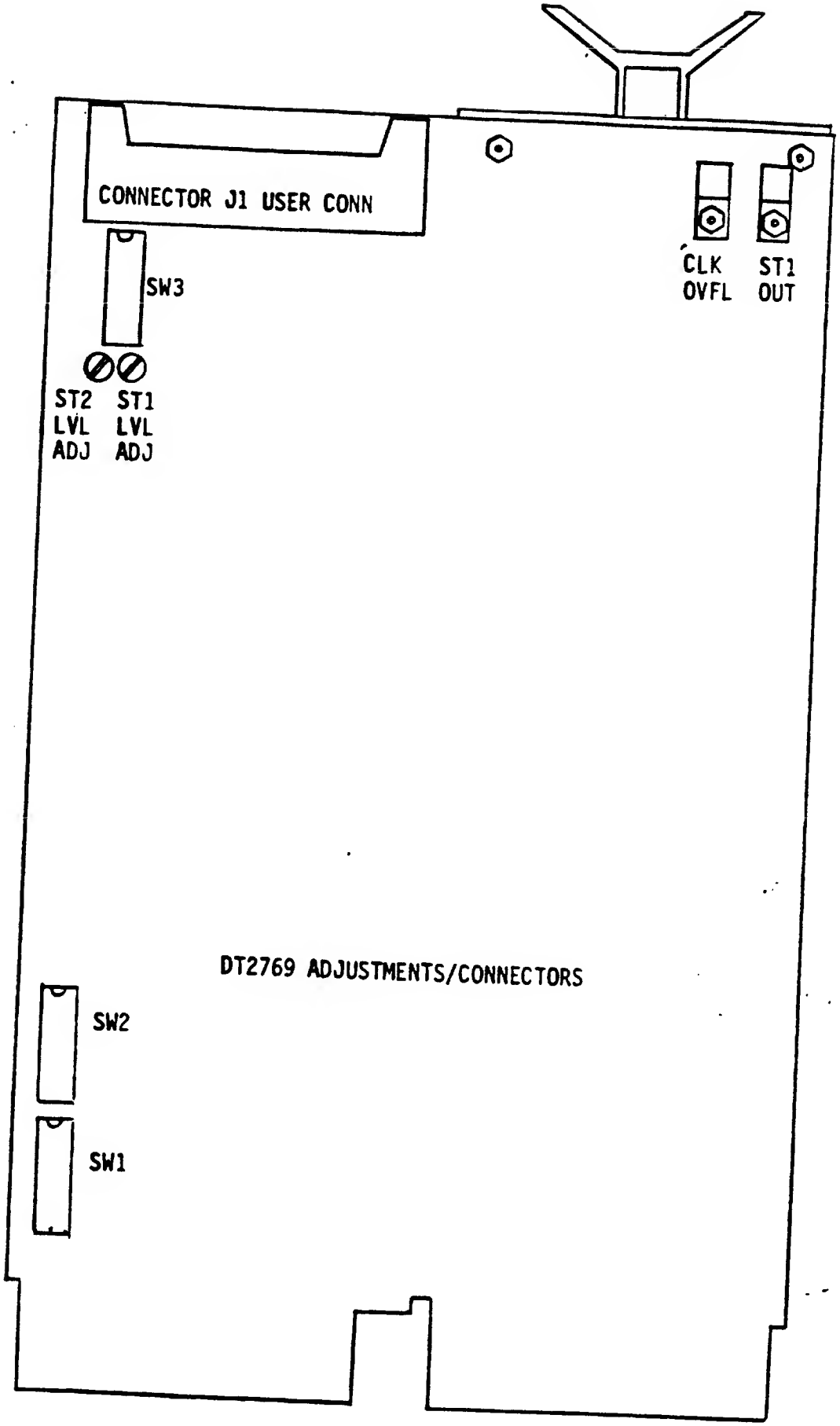
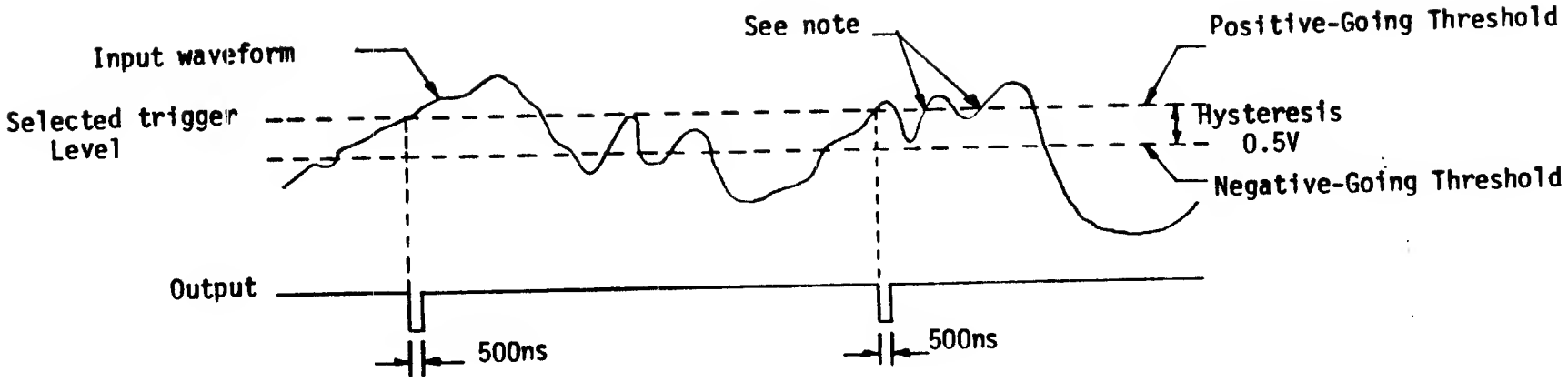

CONNECTOR J1 USER CONN
SW3
ST2 LVL ADJ
ST1 LVL ADJ
CLK OVFL
ST1 OUT
DT2769 ADJUSTMENTS/CONNECTORS
SW2
SW1

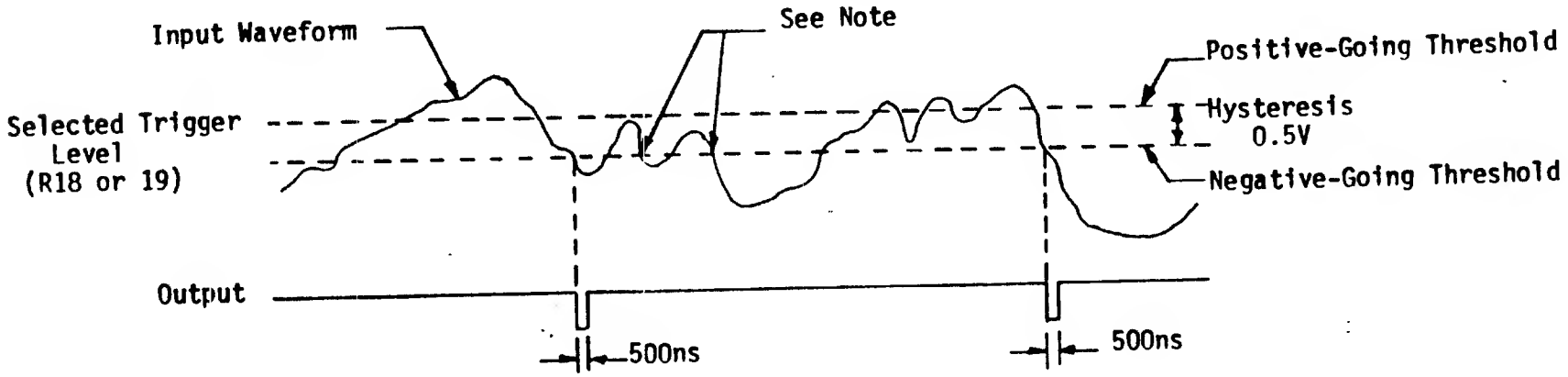
Figure 5.5



NOTE:

ST is retriggered only after input waveform has moved beyond opposite threshold and then again passed selected threshold.

(a) SLOPE SELECTION: SLOPE switch ON (Positive Slope)



NOTE:

ST is retriggered only after input waveform has moved beyond opposite threshold and then again passed selected threshold.

(b) SLOPE SELECTION: SLOPE switch OFF (Negative Slope)

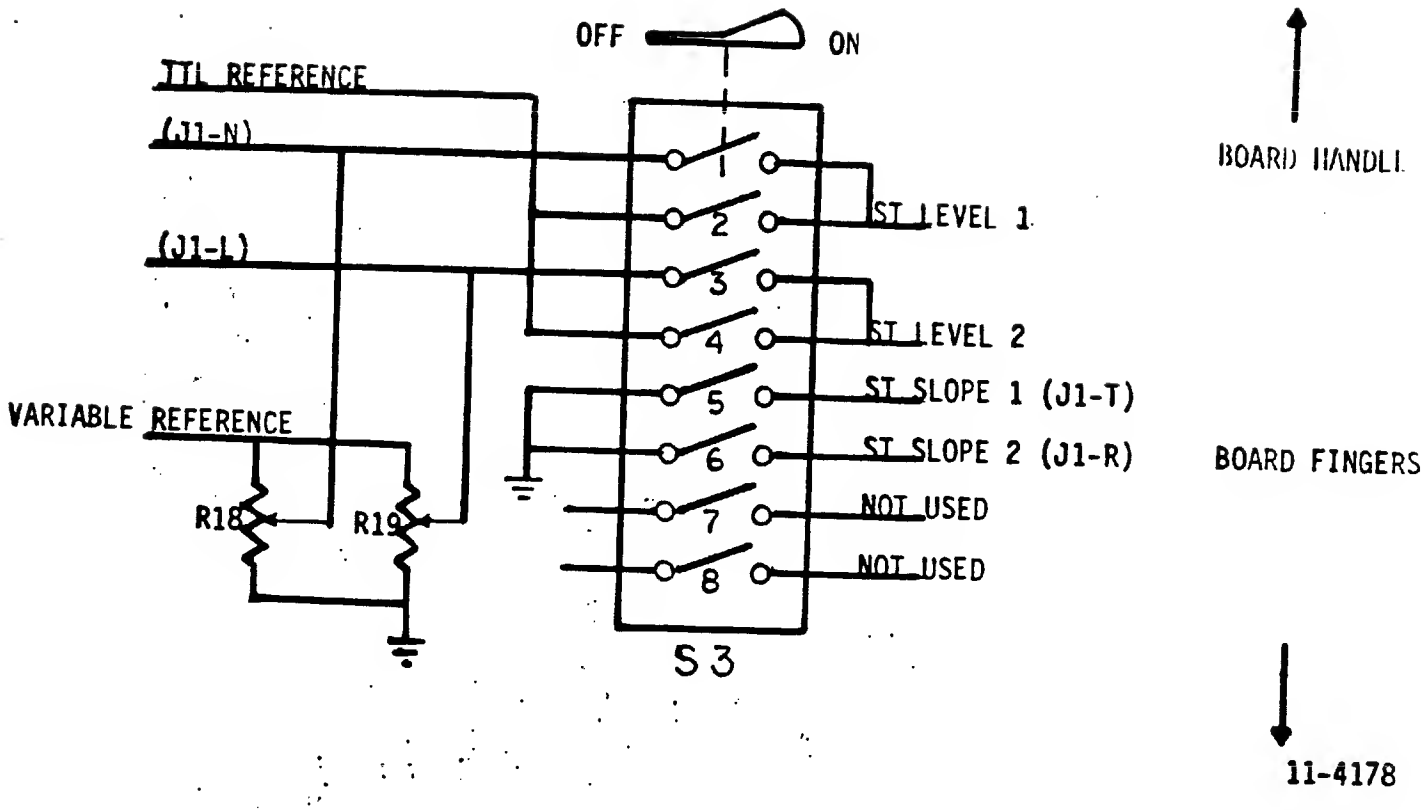


Figure 5.7

SLOPE/REFERENCE LEVEL SELECTOR SWITCHES AND CONTROLS

(NOTE SWITCHES 1 and 2 OR SWITCHES 3 and 4 SHOULD NOT BE ON TOGETHER)

CHAPTER 6

TESTING

6.1 Equipment and System Requirements

In order to assist the user in testing the operation of the DT2769 Real-Time Clock, Data Translation has developed a comprehensive software diagnostic aid designated SP0023. This software is provided in either of the two media: Paper Tape for minimum, paper tape based LSI-11 Systems, or Floppy Disks for more sophisticated RT-11 Systems. The system and test equipment requirements for this software are given below:

SP0023 System Requirements

Paper Tape:
KD11-F (LSI-11) processor, ECO #10 or greater

or

KD11-HA (LSI-11/2) processor

Minimum of 4K words RAM
Serial Interface at Standard DEC console address
Paper tape reader
Data Translation DT2769 Real-Time Clock

Floppy Disk:
KD11-F (LSI-11) processor, ECO #10 or greater

or

KD11-HA (LSI-11/2) processor

Minimum of 8K words RAM
System console terminal at standard DEC console address
DEC compatible dual floppy disc drive system
RT-11 operating system (Version 2 or Version 3)
Data Translation DT2769 Real-Time Clock board.

Test Equipment Requirements

10MHz or greater bandwidth oscilloscope.

6.2.1 Loading SP0023 From Paper Tape

SP0023 is supplied in PDP-11 absolute loader format. To load this release into memory the following steps must be taken:

1. Load the LSI-11 absolute loader (see DEC documentation for information on this).
2. Place the paper tape in the paper tape reader.
3. Start the absolute loader at location XXX500 where XXX is determined by the following table:

SYSTEM MEMORY SIZE	XXX
4K	017
8K	037
12K	057

CHAPTER 6
(Continued)

SYSTEM MEMORY SIZE	XXX
16K	077
20K	117
24K	137
28K or greater	157

Following this procedure will cause SP0023 to be loaded into memory and executed.

6.2.2 Loading Floppy Disk

The SP0023 diskette contains an RT-11 memory image file called SP0024.SAV. This is a linked and executable version of SP0023. To load and execute SP0023 the user should boot up RT-11 in the usual way with the system disk in drive Ø. When RT-11 comes up the user should insert the SP0023 disk into drive 1 and type the following command string to the RT-11 monitor;

.RUN DX1:SP0023 (Return)

This will cause SP0023 to be loaded and executed, at this time the SP0023 monitor will have control of the system and RT-11 will be flushed.

6.3 Using SP0023

SP0023 is a stand-alone software diagnostic package that allows the user to test Data Translation's dual-height series of LSI-11 interface cards. SP0023 does not need RT-11 once it has been loaded and in fact it flushes RT-11 from the system after it has loaded. To allow the user to control the testing procedure a monitor has been included in SP0023 that interfaces to the user. When SP0023 is brought up this monitor is automatically entered. On start up the resident monitor prints out the directory of the various Data Translation interface boards that can be test by SP0023 followed by the monitor prompt character ">" (a right angle-bracket). At this point the user should set the desired model number to be tested using the MODEL command, for example

>M (Space) DT2769 (Return)

would set up SP0023 to test the DT2769 Real-Time Clock. The monitor would then invoke all the necessary initialization routines to test this board and confirm the board model by printing out a confirmation message followed by the default base and vector addresses that will be used. These base and vector addresses have been preset at the factory and need not be changed. If for some reason however a change in the base or vector address is required the user can modify locations 542(base address) and 544(vector address) to the new address. In order to easily facilitate this change the SP0023 monitor also has some subset capabilities of ODT. In particular the user can use the slash and back-slash characters to open and modify memory locations just as in ODT. Therefore if the user needs to change any location he should type the address followed by a slash (/) or back-slash (\), the

CHAPTER 6

(Continued)

monitor will then open that location just as in ODT. The user can roll up or down sequentially in memory by using the line feed or carat (^) keys. The SP0023 monitor is reentered from the ODT mode by typing a carriage return. Like ODT a memory location will only be modified if a valid octal number is typed before any of the ODT terminator characters. For example to change the base and vector address of the DT2769 tests one would type:

```
>542/170420 170430 (line-feed)
>000544/000440 400 (return)
```

to change the base address to 170430 and vector address to 400. Note that this change is only temporary and the default addresses will be reloaded if the model number is retyped or of SP0023 is reloaded. In addition to these commands the SP0023 test executive allows the user to ask for the model directory, to start tests, to loop or halt on tests or even to reboot RT-11. The commands available to the user under the SP0023 test executive are listed on the following page.

Example: User wants to run a scope loop on Test 2 because an error is encountered. Type:

```
>LT (Space) 2
```

In this case the program will loop on Test 2 and inhibit error printouts.

If a test is run and no errors occur the test will return to the SP0023 command level without any other messages. If however, an error does occur then the test will print out the test number and error code. The user may look up the meaning of each error code in the programming listings given in Appendix A.

6.4 Test Descriptions

All descriptions of the tests and the set up requirements needed for any particular test are described in detail in the program listings at the beginning of each test.

6.5 SP0023 Program Description

SP0023 consists of two groups of tests. The first group contains 24_8 tests which test all the logic of the interface board. These tests contain scope loops for debug purposes and will provide an error message if a problem exists. The second set of tests, test the operation of the Schmitt Triggers and require some special interconnections before each test is run, consult the test descriptions preceeding each of these tests to determine the required external set up.

A listing of all the tests for testing DT2769 can be found at the beginning of the program listings.

Note, that the user can run all the logic test (1-24) by using the monitor ALL command. Tests 25-28 however must be run individually.

COMMAND	FUNCTION
ALL	Runs all logic tests that are present for the current device. Generates an error if there is no current device.
BOOT	Jumps to the standard hardware bootstrap (173000). Generates an error if there is no bootstrap present.
DIRECTORY	Displays the contents of the current directory. Generates an error if there is no current directory.
EXIT	Halts the processor
MODEL (space)	Displays the parameters associated with the current device. Generates an error if there is no current device.
MODEL (space) DTXXXX	Searches the current directory for the given model number. If found, makes that model the current device. Generates an error of there is no current directory or if the model number can not be found.
TEST (space)	Runs the last test executed
TEST (space) ⟨number⟩	Runs the indicated test

Test Command prefixes - The following command prefixes are to be used with the TEST command to control the execution of the various tests.

COMMAND PREFIXES TO TEST EXECUTION COMMANDS

R (TEST command only)	repeat this test continuously
L	loop on this test if an error is detected
H	halt test stream if an error is detected
I	inhibit error printout

A control C will terminate any test.

CHAPTER 6
(Continued)

6.6 Requirements to Run Diagnostics

Please note that when SP0023 is executing the Line Time Clock LTC must be turned on (if available) in order that Test #11 execute properly. If a LTC is not present or is turned of Test #11 will report an error that the clock failed to count at the LTC rate (error code 17 mode = 71).

The user should also avoid using DT2769 on an extender card because in certain backplane configurations the Bus Logic Chips do not function properly (due to additional capacitive and inductive effects) and the board will not operate properly.

Note: When running the diagnostics (SP0023), the user must make sure that all the Schmitt Trigger control switches (near connector) are in the "OFF" or open position. If the diagnostic is run with any or all switches in the closed or "ON" position the timing of the board will change due to events occurring at different edges which in turn will cause the diagnostics to falsely report a variety of errors. After having checked out the board with the diagnostics, the user can then set the Schmitt Trigger control switches as required and program the operation as required.

APPENDIX A

SP0023 SOFTWARE LISTINGS

APPENDIX B

DT2769 CIRCUIT SCHEMATICS

```
                        .LIST   TTM
                        .ENABL  LC
                        .TITLE  DT2769  TST-11 MODULE
                        .IDENT  /V02.00/
                        .PSECT  DT2769
000000
                .NLIST  BIN
                ;
                .SBTTL  GENERAL INFORMATION
                ;
                ;
                ; COPYRIGHT (C) 1979, DATA TRANSLATION INCORPORATED. ALL
                ; RIGHTS RESERVED. NO PART OF THIS PROGRAM OR PUBLICATION
                ; MAY BE REPRODUCED WITHOUT THE PRIOR WRITTEN PERMISSION
                ; OF  DATA TRANSLATION INCORPORATED,  4 STRATHMORE ROAD,
                ; NATICK, MASS. 01760.
                ;
                ;
                ; THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
                ; WITHOUT  NOTICE  AND  SHOULD  NOT  BE  CONSTRUED AS A
                ; COMMITMENT BY DATA TRANSLATION INCORPORATED.
                ;
                ;
                ; DATA TRANSLATION CANNOT ASSUME ANY RESPONSIBILITY FOR
                ; THE USE OF ANY PORTION OF THIS SOFTWARE FOR OTHER THAN
                ; ITS INTENDED  DIAGNOSTIC  PURPOSE  IN CALIBRATING AND
                ; TESTING  DATA  TRANSLATION  MANUFACTURED  ANALOG  AND
                ; DIGITAL INTERFACE BOARDS.
                ;
                ;
                ; VERSION 02-00
                ;
                ; JAWED WAHID   6-SEP-78
                ; EDWIN KROEKER 9-JAN-79
                ;
                ;
                ; THIS PROGRAM MODULE CONTAINS ROUTINES TO TEST AND
                ; CALIBRATE DTI MODEL DT2769 REAL TIME CLOCK SUBSYSTEMS
                ; FOR THE LSI-11.  THIS MODULE IS DESIGNED TO OPERATE
                ; UNDER TST-11 SUPERVISION.
                ;
                        .LIST   BIN
```

```
                        .SBTTL  TEST PARAMETER BLOCK (TPB)
                ;
                ; TEST-11 DECLARATION
                ;
                        .MCALL  TST11
000000                  TST11
                ;
                ;
                ;
                ;
                ; TEST PARAMETER BLOCK
                ;
                .NLIST  BIN
        ;
000000  TPB:    .WORD   PARAM           ; ADDRESS OF PARAMETER
                                        ; PRINT-OUT ROUTINE
000002          .BYTE   377             ; RESERVED
000003          .BYTE   30              ; # OF TESTS
        ;
        ; TEST ADDRESS TABLE FOR USE BY TST-11
        ;
000004          .WORD   TEST1,PR7
000010          .WORD   TEST2,PR7
000014          .WORD   TEST3,PR7
000020          .WORD   TEST4,PR7
000024          .WORD   TEST5,PR7
000030          .WORD   TEST6,PR7
000034          .WORD   TEST7,PR7
000040          .WORD   TEST10,PR7
000044          .WORD   TEST11,PR7
000050          .WORD   TEST12,PR7
000054          .WORD   TEST13,PR7
000060          .WORD   TEST14,PR7
000064          .WORD   TEST15,PR7
000070          .WORD   TEST16,PR7
000074          .WORD   TEST17,PR7
000100          .WORD   TEST20,PR7
000104          .WORD   TEST21,PR7
000110          .WORD   TEST22,PR7
000114          .WORD   TEST23,PR7
000120          .WORD   TEST24,PR7
000124          .WORD   TEST25,0
000130          .WORD   TEST26,PR7
000134          .WORD   TEST27,PR7
000140          .WORD   TEST30,PR7
        ;
                        .LIST   BIN
```

```
                        .SBTTL  CSR BIT DEFINITIONS
                ;
                ;
100000  ST2F    =BIT15          ; ST2 FLAG BIT
                ;
040000  INT2    =BIT14          ; ST2 INTERRUPT ENABLE BIT
                ;
020000  ST2G    =BIT13          ; ST2 GO ENABLE BIT
                ;
010000  FOR     =BIT12          ; FLAG OVERRUN BIT.
                ;
004000  DIO     =BIT11          ; DISABLE OSCILLATOR BIT
                ;
002000  MOSC    =BIT10          ; MAINTENANCE OSCILLATOR BIT
                ;
001000  MST2    =BIT9           ; MAINTENANCE ST2 BIT
                ;
000400  MST1    =BIT8           ; MAINTENANCE ST1 BIT
                ;
000200  OVFLO   =BIT7           ; OVERFLOW BIT
                ;
000100  INTOV   =BIT6           ; OVERFLOW INTERRUPT ENABLE BIT
                ;
000001  GO      =BIT0           ; GO BIT
                ;
000000  RATE0   =0
000010  RATE1   =BIT3
000020  RATE2   =BIT4
000030  RATE3   =BIT3+BIT4
000040  RATE4   =BIT5
000050  RATE5   =BIT3+BIT5
000060  RATE6   =BIT4+BIT5
000070  RATE7   =BIT3+BIT4+BIT5
                ;
000000  MODE0   =0
000002  MODE1   =BIT1
000004  MODE2   =BIT2
000006  MODE3   =BIT1+BIT2
```

```
                        .SBTTL  INITIALIZATION
                ;
000144  005037  INIT:   CLR     @#SWR           ; CLEAR SWITCH REGISTER
        000540
000150  012737          MOV     #170420,@#BASE  ; SET UP BASE ADDRESS
        170420
        000542
000156  012737          MOV     #440,@#VECTOR   ; SET UP VECTOR
        000440
        000544
000164  000207          RETURN                  ; ALL DONE
                ;
                ;
                ;
                ;
                        .SBTTL  DISPLAY PARAMETERS
                ;
                ; THIS ROUTINE DISPLAYS THE CURRENT SETTING
                ; OF 'BASE' AND 'VECTOR' ON THE SYSTEM CONSOLE
                ; TERMINAL.
                ;
000166          PARAM:  PRINT   <  BASE ADDRESS = >
000212  013700          MOV     @#BASE,R0       ; GET BASE ADDRESS
        000542
000216                  OCT16                   ; DISPLAY
000220                  CRLF
000222                  PRINT   <VECTOR ADDRESS = >
000246  013700          MOV     @#VECTOR,R0     ; GET VECTOR ADDRESS
        000544
000252                  OCT16                   ; DISPLAY
000254                  CRLF
000256  000207          RETURN                  ; ALL DONE
                ;
                ;
                ;
                ;
                ;
                        .SBTTL  MODEL TESTING INFORMATION
                ;
                ; THIS CODE MODULE CONTAINS THE ROUTINES NECESSARY TO
                ; TEST THE FOLLOWING DTI INTERFACE MODEL:
                ;
                ;       DT2769 REAL TIME CLOCK
                ;
        000144' I2769   ==INIT
        000000' T2769   ==TPB
```

```
                        .SBTTL  ERROR REPORTERS
                ;
                ;
                ; THIS ROUTINE PROVIDES ERROR REPORTING FOR BUS
                ; TIME-OUT ERRORS (NO BRPLY FROM INTERFACE).
                ;
000260          NORPLY: PRINT   <NO BRPLY WHEN ACCESSING LOCATION >
000324  010100          MOV     R1,R0
000326                  OCT16                   ; DISPLAY ADDRESS
000330                  CRLF
000332  000207          RETURN                  ; DONE
                ;
                ;
                ; THIS ROUTINE PROVIDES ERROR REPORTING FOR REGISTER
                ; BIT ERRORS (ONE OR MORE INCORRECT BITS IN A REGISTER).
                ;
000334          REG:    PRINTC  <REGISTER ERROR>
000356                  PRINT   <ADDRESS: >
000372                  PUSH    R0              ; SAVE R0
000374  010100          MOV     R1,R0           ; GET ADDRESS
000376                  OCT16
000400                  CRLF
000402                  PRINT   <EXPECTED:>
000416  010200          MOV     R2,R0           ; GET EXPECTED VALUE
000420                  OCT16                   ; DISPLAY
000422                  CRLF
000424                  PRINT   <FOUND:    >
000440  011600          MOV     (SP),R0         ; GET BAD BITS
000442  074200          XOR     R2,R0           ; GENERATE SNAPSHOT
000444                  OCT16                   ; DISPLAY
000446                  CRLF                    ; FORMATTING
000450                  PRINT   <BITS:     >
000464                  POP     R0              ; GET ERROR BITS
000466                  OCT16                   ; DISPLAY
000470                  CRLF
000472  000207          RETURN                  ; DONE
```

```
                        ; CLOCK FUNCTION ERROR REPORTER
                        ;
                        ; THIS REPORTER HANDLES SOME OF THE CLOCK FUNCTION
                        ; ERRORS (NOT COUNTING, DIVIDER ERRORS, ETC.).
                        ;
000474                  CLOCK:  PRINTC  <CLOCK FUNCTION ERROR>
000524                          PRINT   <LAST CSR: >
000542                          OCT16                   ; DISPLAY R0
000544                          TAB
000546                          PRINT   <RATE,MODE: >
000564  010200                  MOV     R2,R0
000566                          OCT8                    ; DISPLAY R2
000570                          CRLF
000572                          PRINT   <BPR: >
000602  016100                  MOV     2(R1),R0        ; DISPLAY CURRENT BPR
        000002
000606                          OCT16
000610                          CRLF
000612  000207                  RETURN                  ; ALL DONE
```

```
                        .SBTTL  SWITCH PACK S3 CONFIGURATION
                ;
                ; SWITCH PACK S3 MUST HAVE ALL SWITCHES IN THE OFF OR
                ; OPEN POSITION FOR THE FOLLOWING TESTS TO EXECUTE
                ; PROPERLY.  FAILURE TO SET S3 IN THIS STATE WILL
                ; CAUSE EXTRANEOUS ERRORS TO BE REPORTED.
                ;
                ;
                        .SBTTL  TEST 1:  BRPLY FROM ALL REGISTERS
                ;
                ; THIS TEST VERIFIES THAT THE INTERFACE SYSTEM RESPONDS
                ; WITH A BUS REPLY SIGNAL DURING A BUS DATIO BUS CYCLE.
                ; ALL REGISTERS AVAILABLE ON THE BOARD ARE CHECKED.
                ;
                ;
000614  010602  TEST1:  MOV     SP,R2           ; SAVE SP
000616                  RELMOV  #3$,R0          ; SET UP TRAP TO 4
000624  010037          MOV     R0,@#4
        000004
000630  013701  2$:     MOV     @#BASE,R1       ; GET ADDRESS
        000542
000634                  SCOPE                   ; DECLARE LOOP POINT
000636  005011          CLR     (R1)            ; DATIO BUS CYCLE
000640  062701          ADD     #2,R1           ; NEXT REGISTER
        000002
000644                  SCOPE                   ; DECLARE LOOP POINT
000646  005011          CLR     (R1)            ; DATIO BUS CYCLE
000650                  EXIT
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 1 - BUS TIMEOUT
                ;
                ; ********************************************************
                ;
000652  011603  3$:     MOV     (SP),R3         ; GET OFFENDING PC
000654  010206          MOV     R2,SP           ; RESTORE STACK
000656                  ERROR   1,NORPLY        ; REPORT ERROR
000662  000113          JMP     (R3)            ; CONTINUE TEST
```

```
                        .SBTTL  TEST 2:  CHECK CSR R/W BITS
                ;
                ; THIS TEST CHECKS THE READ/WRITE BITS IN THE CSR
                ; BITS ARE CHECKED FOR BOTH SET AND RESET CAPABILITY
                ;
000664  013701  TEST2:  MOV     @#BASE,R1       ; GET ADDRESS
        000542
000670                  RELMOV  #CSRBIT,R4      ; GET BIT TABLE ADDRESS
000676  012705         MOV     #12,R5          ; NUMBER OF BITS
        000012
000702  012402  1$:     MOV     (R4)+,R2        ; GET BIT
000704  010203         MOV     R2,R3           ; GENERATE MASK
000706  005103         COM     R3
000710                  SCOPE                   ; DECLARE LOOP POINT
000712  010211         MOV     R2,(R1)         ; SET BIT
000714  011100         MOV     (R1),R0         ; GET BIT
000716  040300         BIC     R3,R0           ; TEST BIT
000720  001002         BNE     2$              ; SET - SKIP ERROR
                ;
                ; ***************************************************
                ;
                ;       ERROR CODE 2 - BIT SET ERROR, CSR
                ;
                ; ***************************************************
                ;
000722                  ERROR   2,REG           ; REPORT ERROR
                ;
000726  005002  2$:     CLR     R2              ; INIT. TEST REGISTER
000730                  SCOPE                   ; DECLARE LOOP POINT
000732  010211         MOV     R2,(R1)         ; CLEAR BIT
000734  011100         MOV     (R1),R0         ; GET BIT
000736  040300         BIC     R3,R0           ; TEST BIT
000740  001402         BEQ     3$              ; CLEAR - SKIP ERROR
                ;
                ; ***************************************************
                ;
                ;       ERROR CODE 3 - BIT CLEAR ERROR, CSR
                ;
                ; ***************************************************
                ;
000742                  ERROR   3,REG           ; REPORT ERROR
                ;
000746  077523  3$:     SOB     R5,1$           ; LOOP UNTIL DONE
000750                  EXIT                    ; ALL DONE
                ;
                ;
                .NLIST  BIN
000752  CSRBIT: .WORD   GO,BIT1,BIT2,BIT3,BIT4,BIT5
000766          .WORD   INTOV,DIO,ST2G,INT2
                        .LIST   BIN
```

```
                        .SBTTL  TEST 3:  BYTE OPERATION OF CSR
                ;
                ; THIS TEST VERIFIES HIGH AND LOW BYTE OPERATIONS
                ; INVOLVING THE CSR.
                ;
                ;
001000 013701   TEST3:  MOV     @#BASE,R1       ; GET ADDRESS
       000542
001004 005011           CLR     (R1)            ; CLEAR CSR
001006 005002           CLR     R2              ; INIT. TEST REGISTER
001010                  SCOPE                   ; DECLARE LOOP POINT
001012 112711           MOVB    #-1,(R1)        ; SET R/W BITS
       177777
001016 011100           MOV     (R1),R0         ; GET CSR AS WORD
001020 042700           BIC     #<ST2F+377>,R0  ; IGNORE LOW BYTE
       100377
001026 074200           XOR     R2,R0           ; TEST BITS
001030 001402           BEQ     1$              ; OK - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 4 - HIGH BYTE LOADED DURING
                ;                      A LOW BYTE OPERATION
                ;
                ; ********************************************************
                ;
001032                  ERROR   4,REG           ; REPORT ERROR
                ;
001036 005011   1$:     CLR     (R1)            ; CLEAR CSR
001040 005002           CLR     R2              ; INIT. TEST REGISTER
001042 005201           INC     R1              ; POINT TO HIGH BYTE
001044                  SCOPE                   ; DECLARE LOOP POINT
001046 112711           MOVB    #-1,(R1)        ; SET R/W BITS
       177777
001052 016100           MOV     -1(R1),R0       ; GET CSR AS WORD
       177777
001056 042700           BIC     #<177400+OVFLO+GO>,R0
       177601
001062 074200           XOR     R2,R0           ; TEST BITS
001064 001402           BEQ     2$              ; OK - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 5 - LOW BYTE LOADED DURING
                ;                      A HIGH BYTE OPERATION
                ;
                ; ********************************************************
                ;
001066                  ERROR   5,REG           ; REPORT ERROR
                ;
001072          2$:     EXIT
```

```
                        .SBTTL  TEST 4:  BUFFER PRESET REGISTER BITS
                ;
                ; THIS TEST VERIFIES THE OPERATION OF THE BUFFER PRESET
                ; REGISTER UNDER ALL ALLOWABLE BIT COMBINATIONS
                ;
001070  013701  TEST4:  MOV     @#BASE,R1       ; GET ADDRESS
        000542
001074  005011          CLR     (R1)            ; SET MODE 0
001076  062701          ADD     #2,R1           ; BUMP POINTER
        000002
001102  005002          CLR     R2              ; INIT. TEST REGISTER
001104  005003          CLR     R3              ; ITERATION COUNT
001106                  SCOPE                   ; DECLARE LOOP POINT
001110  010211  1$:     MOV     R2,(R1)         ; SET BITS
001112  011100          MOV     (R1),R0         ; GET BITS
001114  074200          XOR     R2,R0           ; TEST BITS
001116  001402          BEQ     2$              ; OKAY - NEXT STATE
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 6 - BIT ERROR, BPR
                ;
                ; ********************************************************
                ;
001120                  ERROR   6,REG           ; REPORT ERROR
                ;
001124  005202  2$:     INC     R2              ; NEXT STATE
001126  077310          SOB     R3,1$           ; LOOP UNTIL DONE
001130                  EXIT
```

```
                        .SBTTL  TEST 5:  BINITL ACTION
                ;
                ; THIS TEST VERIFIES THAT THE BINITL SIGNAL CLEARS
                ; THE PROPER CSR AND BPR BITS.
                ;
                ;
001132  013701  TEST5:  MOV     @#BASE,R1       ; GET ADDRESS
        000542
001136  005002          CLR     R2              ; INIT TEST REGISTER
001140                  SCOPE                   ; DECLARE LOOP POINT
001142  012711          MOV     #-1,(R1)        ; SET BITS
        177777
001146  000005          RESET                   ; CLEAR BITS
001150  011100          MOV     (R1),R0         ; GET BITS
001152  001402          BEQ     1$              ; OKAY - ALL ZERO
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 7 - BINITL DOES NOT CLEAR
                ;                      CSR BITS
                ;
                ; ********************************************************
                ;
001154                  ERROR   7,REG           ; REPORT ERROR
                ;
001160  062701  1$:     ADD     #2,R1           ; BUMP POINTER
        000002
001164                  SCOPE                   ; DECLARE LOOP POINT
001166  012711          MOV     #-1,(R1)        ; SET BITS
        177777
001172  000005          RESET                   ; CLEAR BITS
001174  011100          MOV     (R1),R0         ; GET BITS
001176  001402          BEQ     2$              ; OKAY - ALL ZERO
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 10 - BINITL DOES NOT CLEAR
                ;                       BPR BITS
                ;
                ; ********************************************************
                ;
001200                  ERROR   10,REG          ; REPORT ERROR
                ;
001204          2$:     EXIT                    ; ALL DONE
```

```
                        .SBTTL  TEST 6:  CLOCK COUNT REGISTER
                ;
                ; THIS TEST VERIFIES THAT THE BUFFER PRESET REGISTER
                ; CAN BE PROPERLY TRANSFERRED TO THE CLOCK COUNT
                ; REGISTER.
                ;
001206  013701  TEST6:  MOV     @#BASE,R1       ; GET ADDRESS
        000542
001212  010103          MOV     R1,R3           ; GENERATE BPR ADDRESS
001214  062703          ADD     #2,R3
        000002
001220  005004          CLR     R4              ; ITERATION COUNT
001222  005002          CLR     R2              ; INIT TEST REGISTER
001224                  SCOPE                   ; DECLARE LOOP POINT
001226  005011  1$:     CLR     (R1)            ; SET MODE 0
001230  010213          MOV     R2,(R3)         ; LOAD BPR
001232  052711          BIS     #GO,(R1)        ; TRANSFER TO CCR
        000001
001236  052711          BIS     #<DIO+RATE0+MODE2+GO>,(R1)
        004005
001242  052711          BIS     #MST2,(R1)
        001000
001246  011300          MOV     (R3),R0         ; GET CLOCK COUNT
001250  074200          XOR     R2,R0           ; CHECK
001252  001402          BEQ     2$              ; OKAY- SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 11 - BIT ERROR, CLOCK COUNT
                ;
                ; ********************************************************
                ;
001254                  ERROR   11,REG          ; REPORT ERROR
                ;
001260  005202  2$:     INC     R2              ; NEXT STATE
001262  077417          SOB     R4,1$           ; LOOP BACK
001264                  EXIT
```

```
                        .SBTTL  TEST 7:   CLOCK STATE TRANSITIONS
                ;
                ; THIS TEST, USING MAINT ST1, VERIFIES THAT THE CLOCK
                ; CAN BE INCREMENTED PROPERLY.
                ;
001266  013701  TEST7:  MOV     @#BASE,R1       ; GET BASE ADDRESS
        000542
001272  012702          MOV     #1,R2           ; INIT TEST REGISTERS
        000001
001276  005003          CLR     R3
001300                  SCOPE                   ; DECLARE LOOP POINT
001302  005011  1$:     CLR     (R1)            ; CLEAR CSR
001304  010361          MOV     R3,2(R1)        ; LOAD BPR
        000002
001310  012711          MOV     #<RATE6+MODE0+GO>,(R1)
        000061
001314  052711          BIS     #MST1,(R1)      ; GENERATE ST1
        000400
001320  052711          BIS     #<DIO+RATE0+MODE2+GO>,(R1)
        004005
001324  052711          BIS     #MST2,(R1)      ; CLOCK TO BPR
        001000
001330  016100          MOV     2(R1),R0        ; GOT IT
        000002
001334  074200          XOR     R2,R0           ; CHECK IT
001336  001402          BEQ     2$              ; OKAY - NEXT STATE
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 12 - CLOCK STATE TRANSITION ERROR
                ;
                ; **************************************************
                ;
001340                  ERROR   12,REG          ; REPORT ERROR
                ;
001344  005202  2$:     INC     R2              ; NEXT STATE
001346  005203          INC     R3
001350  001354          BNE     1$              ; IF ZERO THEN DONE
001352                  EXIT
```

```
                        .SBTTL  TEST 10: OVERFLOW BIT
                ;
                ; THIS TEST VERIFIES THAT THE OVERFLOW BIT FUNCTIONS
                ; PROPERLY IN MODES 0 AND 1.
                ;
001354  013701  TEST10: MOV     @#BASE,R1       ; GET ADDRESS
        000542
001360                  SCOPE                   ; DECLARE LOOP POINT
001362  005011          CLR     (R1)            ; CLEAR CSR
001364  012761          MOV     #-1,2(R1)       ; LOAD BPR
        177777
        000002
001372  012711          MOV     #<RATE6+MODE0+GO>,(R1)
        000061
001376  052711          BIS     #MST1,(R1)      ; GENERATE ST1
        000400
001402  105711          TSTB    (R1)            ; OVERFLOW SET?
001404  100401          BMI     1$              ; YES - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 13 - OVERFLOW FLAG NOT SET
                ;
                ; ********************************************************
                ;
001406                  ERROR   13              ; REPORT ERROR
                ;
001410  032711  1$:     BIT     #GO,(R1)        ; GO BIT CLEARED?
        000001
001414  001401          BEQ     2$              ; YES - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 14 - GO BIT NOT CLEARED BY
                ;                       OVERFLOW IN MODE 0
                ;
                ; ********************************************************
                ;
001416                  ERROR   14              ; REPORT ERROR
```

```
 1 001420             2$:     SCOPE                           ; DECLARE LOOP POINT
 2 001422  005011             CLR     (R1)                    ; CLEAR CSR
 3 001424  012761             MOV     #-1,2(R1)               ; LOAD BPR
           177777
           000002
 4 001432  012711             MOV     #<RATE6+MODE1+GO>,(R1)
           000063
 5 001436  052711             BIS     #MST1,(R1)              ; GENERATE ST1
           000400
 6 001442  105711             TSTB    (R1)                    ; OVERFLOW FLAG SET?
 7 001444  100401             BMI     3$                      ; YES - SKIP ERROR
 8                       ;
 9                       ; ***************************************************
10                       ;
11                       ;       ERROR CODE 15 - OVERFLOW FLAG NOT SET
12                       ;
13                       ; ***************************************************
14                       ;
15 001446                    ERROR   15                      ; REPORT ERROR
16                       ;
17 001450  032711        3$:     BIT     #GO,(R1)            ; GO BIT CLEARED?
           000001
18 001454  001001             BNE     4$                      ; NO - SKIP ERROR
19                       ;
20                       ; ***************************************************
21                       ;
22                       ;       ERROR CODE 16 - GO BIT CLEARED BY
23                       ;                       OVERFLOW IN MODE 1
24                       ;
25                       ; ***************************************************
26                       ;
27 001456                    ERROR   16                      ; REPORT ERROR
28                       ;
29 001460                4$:     EXIT                        ; ALL DONE
```

```
                        .SBTTL  TEST 11: CLOCK COUNTING, MODE 0
                ;
                ; THIS TEST VEIFIES THAT THE CLOCK COUNTS AT THE 1MHZ,
                ; 100 KHZ, 10KHZ, 1KHZ, 100HZ, AND BEVENT RATES.
                ;
001462  013701  TEST11: MOV     @#BASE,R1       ; GET ADDRESS
        000542
001466  012704          MOV     #6,R4           ; ITERATION COUNT
        000006
001472  012702          MOV     #<RATE1+MODE0+GO>,R2
        000011
001476  010203  1$:     MOV     R2,R3           ; GENERATE SPECIAL CSR
001500  052703          BIS     #<DIO+RATE0+MODE2+GO>,R3
        004005
001504                  SCOPE                   ; DECLARE LOOP POINT
001506  005011          CLR     (R1)            ; CLEAR CSR
001510  005061          CLR     2(R1)           ; CLEAR BPR
        000002
001514  010211          MOV     R2,(R1)         ; SET RATE
001516  005000          CLR     R0              ; DELAY
001520  077001  2$:     SOB     R0,2$
001522  011100          MOV     (R1),R0         ; SAVE CSR
001524  010311          MOV     R3,(R1)         ; READ CLOCK
001526  052711          BIS     #MST2,(R1)      ; AFTER ST2
        001000
001532  005761          TST     2(R1)           ; DID IT COUNT?
        000002
001536  001004          BNE     3$              ; YES - CONTINUE
001540  105700          TSTB    R0              ; PERHAPS THERE WAS
001542  100402          BMI     3$              ; OVERFLOW
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 17 - CLOCK DIDN'T COUNT
                ;
                ; ********************************************************
                ;
001544                  ERROR   17,CLOCK        ; REPORT ERROR
                ;
001550  062702  3$:     ADD     #10,R2          ; GENERATE NEXT STATE
        000010
001554  022702          CMP     #<RATE6+GO>,R2  ; SKIP ST1 RATE
        000061
001560  001773          BEQ     3$
001562  077433          SOB     R4,1$           ; LOOP UNTIL DONE
001564                  EXIT
```

```
                        .SBTTL  TEST 12: CLOCK COUNTING, MODE 0
                ;
                ; THIS TEST VERIFIES THAT THE CLOCK DOES NOT COUNT WHEN
                ; THE STOP RATE IS SELECTED.
                ;
001566  013701  TEST12: MOV     @#BASE,R1       ; GET ADDRESS
        000542
001572                  SCOPE                   ; DECLARE LOOP POINT
001574  005011          CLR     (R1)            ; CLEAR CSR
001576  005061          CLR     2(R1)           ; CLEAR BPR
        000002
001602  012711          MOV     #<RATE0+MODE0+GO>,(R1)
        000001
001606  052711          BIS     #MST1,(R1)      ; GENERATE ST1
        000400
001612  005000          CLR     R0              ; WAIT
001614  077001  1$:     SOB     R0,1$
001616  011100          MOV     (R1),R0         ; SAVE CSR
001620  012711          MOV     #<DIO+RATE0+MODE2+GO>,(R1)
        004005
001624  052711          BIS     #MST2,(R1)      ; GENERATE ST2
        001000
001630  005761          TST     2(R1)           ; DID IT COUNT?
        000002
001634  001002          BNE     2$              ; YES - ERROR
001636  105700          TSTB    R0              ; OVERFLOW?
001640  100002          BPL     3$              ; NO - SKIP ERROR
                ;
                ; ***************************************************
                ;
                ;       ERROR CODE 21 - CLOCK COUNTED WHEN STOP
                ;                       RATE WAS SELECTED
                ;
                ; ***************************************************
                ;
001642          2$:     ERROR   21,CLOCK        ; REPORT ERROR
                ;
001646          3$:     EXIT
```

```
                        .SBTTL  TEST 13: CLOCK COUNTING, MODE 1
                ;
                ; THIS TEST VERIFIES THAT THE CLOCK WILL COUNT IN
                ; MODE 1.
                ;
001650  013701  TEST13: MOV     @#BASE,R1       ; GET ADDRESS
        000542
001654                  SCOPE                   ; DECLARE LOOP POINT
001656  005011          CLR     (R1)            ; CLEAR CSR
001660  012761          MOV     #-1,2(R1)       ; LOAD BPR.
        177777
        000002
001666  012711          MOV     #<RATE1+MODE1+GO>,(R1)
        000013
001672  005000          CLR     R0              ; WAIT
001674  077001  1$:     SOB     R0,1$
001676  105711          TSTB    (R1)            ; OVERFLOW?
001700  100402          BMI     2$              ; YES - SKIP ERROR
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 22 - CLOCK DIDN'T OVERFLOW
                ;
                ; **************************************************
                ;
001702                  ERROR   22,CLOCK        ; REPORT ERROR
                ;
001706          2$:     EXIT
```

```
                        .SBTTL  TEST 14: MAINTENANCE ST2
                ;
                ; THIS TEST VERIFIES THAT BITS 15 AND 0 CAN BE
                ; PROPERLY EXERCISED BY THE MAINTENANCE ST2 BIT.
                ;
001710  013701  TEST14: MOV     @#BASE,R1       ; GET ADDRESS
        000542
001714  005011          CLR     (R1)            ; CLEAR CSR
001716  012702          MOV     #<ST2F+GO>,R2   ; INIT. TEST REGISTER
        100001
001722                  SCOPE                   ; DECLARE LOOP POINT
001724  012711          MOV     #GO,(R1)                ; SET GO BIT
        000001
001730  052711          BIS     #MST2,(R1)      ; GENERATE ST2
        001000
001734  011100          MOV     (R1),R0         ; GET CSR
001736  074200          XOR     R2,R0           ; TEST BITS
001740  001402          BEQ     1$              ; GOOD - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 23 - ST2 FAILED TO SET BIT 15
                ;
                ; ********************************************************
                ;
001742                  ERROR   23,REG          ; REPORT ERROR
                ;
001746  005011  1$:     CLR     (R1)            ; CLEAR CSR
001750  012702          MOV     #<ST2F+GO>,R2
        100001
001754                  SCOPE                   ; DECLARE LOOP POINT
001756  012711          MOV     #ST2G,(R1)      ; ST2 GO ENB
        020000
001762  052711          BIS     #MST2,(R1)      ; GENERATE ST2
        001000
001766  011100          MOV     (R1),R0         ; GET CSR
001770  074200          XOR     R2,R0           ; TEST BITS
001772  001402          BEQ     2$              ; GOOD - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 24 - ST2 FAILED TO SET GO BIT
                ;                       AND/OR CLEAR ST2 GO ENABLE
                ;
                ; ********************************************************
                ;
001774                  ERROR   24,REG          ; REPORT ERROR
                ;
002000          2$:     EXIT
```

```
                        .SBTTL  TEST 15: INTERRUPT ON OVERFLOW
                ;
                ; THIS TEST VERIFIES THAT THE CLOCK CAN INTERRUPT THE
                ; PROCESSOR PROPERLY ON OVERFLOW
                ;
002002  013701  TEST15: MOV     @#BASE,R1       ; GET ADDRESS
        000542
002006  005011          CLR     (R1)            ; CLEAR CSR
002010  013703          MOV     @#VECTOR,R3     ; GET VECTOR ADDRESS
        000544
002014                  RELMOV  #2$,R0          ; GENERATE ISR ADDRESS
002022  010013          MOV     R0,(R3)         ; SAVE
002024  005004          CLR     R4
002026  012705          MOV     #PR7,R5
        000340
002032                  SCOPE                   ; DECLARE LOOP POINT
002034  012761          MOV     #-1,2(R1)       ; LOAD BPR
        177777
        000002
002042  012711          MOV     #<INTOV+RATE6+MODE0+GO>,(R1)
        000161
002046  052711          BIS     #MST1,(R1)      ; GENERATE ST1
        000400
002052  105711          TSTB    (R1)            ; OVERFLOW SET?
002054  100402          BMI     1$              ; YES - CONTINUE
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 25 - OVERFLOW BIT NOT SET
                ;
                ; ********************************************************
                ;
002056                  ERROR   25              ; REPORT ERROR
002060  000410          BR      3$              ; CAN'T CONTINUE
                ;
002062  106404  1$:     MTPS    R4              ; ENABLE INTERRUPTS
002064  000240          NOP                     ; WINDOW
002066  106405          MTPS    R5              ; DISABLE INTERRUPTS
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 26 - NO INTERRUPT ON OVERFLOW
                ;
                ; ********************************************************
                ;
002070                  ERROR   26              ; REPORT ERROR
002072  000403          BR      3$              ; CAN'T CONTINUE
                ;
002074  062706  2$:     ADD     #4,SP           ; RESTORE STACK
        000004
002100  005011          CLR     (R1)            ; CLEAR CSR
002102          3$:     EXIT
```

```
                        .SBTTL  TEST 16: INTERRUPT ON ST2
                ;
                ; THIS TEST VERIFIES THAT THE CLOCK CAN INTERRUPT THE
                ; PROCESSOR PROPERLY ON AN ST2 PULSE.
                ;
002104  013701  TEST16: MOV     @#BASE,R1       ; GET ADDRESS
        000542
002110  005011          CLR     (R1)            ; CLEAR CSR
002112  013703          MOV     @#VECTOR,R3     ; GET VECTOR ADDRESS
        000544
002116  062703          ADD     #4,R3           ; ADJUST VECTOR
        000004
002122                  RELMOV  #2$,R0          ; GENERATE ISR ADDRESS
002130  010013          MOV     R0,(R3)         ; SAVE
002132  005004          CLR     R4
002134  012705          MOV     #PR7,R5
        000340
002140                  SCOPE                   ; DECLARE LOOP POINT
002142  012711          MOV     #<INT2+GO>,(R1) ; LOAD CSR
        040001
002146  052711          BIS     #MST2,(R1)      ; GENERATE ST2
        001000
002152  005711          TST     (R1)            ; ST2 SET?
002154  100402          BMI     1$              ; YES - CONTINUE
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 27 - ST2 NOT SET
                ;
                ; ********************************************************
                ;
002156                  ERROR   27              ; REPORT ERROR
002160  000410          BR      3$              ; CAN'T CONTINUE
                ;
002162  106404  1$:     MTPS    R4              ; ENABLE INTERRUPTS
002164  000240          NOP                     ; WINDOW
002166  106405          MTPS    R5              ; DISABLE INTERRUPTS
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 30 - NO INTERRUPT ON ST2
                ;
                ; ********************************************************
                ;
002170                  ERROR   30              ; REPORT ERROR
002172  000403          BR      3$              ; CAN'T CONTINUE
                ;
002174  062706  2$:     ADD     #4,SP           ; RESTORE STACK
        000004
002200  005011          CLR     (R1)            ; CLEAR CSR
002202          3$:     EXIT
```

```
                        .SBTTL  TEST 17: FLAG OVERRUN BIT
                ;
                ; THIS TEST VERIFIES THAT THE FOR BIT IN THE CSR
                ; OPERATES PROPERLY.
                ;
002204  013701  TEST17: MOV     @#BASE,R1       ; GET ADDRESS
        000542
002210                  SCOPE                   ; DECLARE LOOP POINT
002212  005011          CLR     (R1)            ; CLEAR CSR
002214  012711          MOV     #GO,(R1)        ; SET GO BIT
        000001
002220  052711          BIS     #MST2,(R1)      ; GENERATE ST2
        001000
002224  052711          BIS     #MST2,(R1)      ; GENERATE ANOTHER
        001000
002230  032711          BIT     #<FOR+ST2F>,(R1)
        110000
002234  001001          BNE     1$              ; YES - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 31 - FLAG OVERRUN BIT/ST2 FLAG
                ;                       NOT SET AS EXPECTED
                ;
                ; ********************************************************
                ;
002236                  ERROR   31              ; REPORT ERROR
                ;
002240  042711  1$:     BIC     #GO,(R1)        ; CLEAR GO BIT
        000001
002244  052711          BIS     #GO,(R1)        ; SET GO BIT
        000001
002250  032711          BIT     #<FOR+ST2F>,(R1)
        110000
002254  001401          BEQ     2$              ; GOOD- SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 32 - FLAG OVERRUN BIT/ST2 FLAG
                ;                       NOT CLEARED BY GO BIT
                ;
                ; ********************************************************
                ;
002256                  ERROR   32              ; REPORT ERROR
```

TEST 17: FLAG OVERRUN BIT

```
                        2$:     SCOPE                   ; DECLARE LOOP POINT
 1 002260                       CLR     (R1)            ; CLEAR CSR
 2 002262  005011               MOV     #-1,2(R1)       ; LOAD BPR
 3 002264  012761
           177777
           000002               MOV     #<RATE6+MODE1+GO>,(R1)
 4 002272  012711               BIS     #MST1,(R1)      ; GENERATE ST1
           000063
 5 002276  052711               TSTB    (R1)            ; OVERFLOW SET?
           000400               BMI     3$              ; YES - CONTINUE
 6 002302  105711
 7 002304  100402
 8                      ;*****************************************************
 9                      ;
10                      ;       ERROR CODE 33 - OVERFLOW BIT NOT SET
11                      ;
12                      ;*****************************************************
13                      ;
14                              ERROR   33              ; REPORT ERROR
15 002306                       BR      5$              ; CAN'T CONTINUE
16 002310  000412
17                      ;
18 002312  032711       3$:     BIT     #FOR,(R1)       ; FOR BIT SET?
           010000               BEQ     4$              ; NO - CONTINUE
19 002316  001401
20                      ;*****************************************************
21                      ;
22                      ;       ERROR CODE 34 - FLAG OVERRUN BIT SET WHEN
23                      ;                       IT SHOULDN'T BE
24                      ;
25                      ;*****************************************************
26                      ;
27                              ERROR   34              ; REPORT ERROR
28 002320
29                      ;
30 002322  052711       4$:     BIS     #MST1,(R1)      ; GENERATE ST1
           000400               BIT     #FOR,(R1)       ; FOR SET?
31 002326  032711               BNE     5$              ; YES - SKIP ERROR
           010000
32 002332  001001
33                      ;*****************************************************
34                      ;
35                      ;       ERROR CODE 35 - FLAG OVERRUN BIT NOT SET
36                      ;
37                      ;*****************************************************
38                      ;
39                              ERROR   35              ; REPORT ERROR
40 002334
41                      ;
42 002336               5$:     EXIT
```

```
                        .SBTTL  TEST 20: OSCILLATOR
                ;
                ; THIS TEST VERIFIES THAT CSR BIT 11 DISABLES THE
                ; INTERNAL OSCILLATOR, AND THAT THE MAINTENANCE
                ; OSCILLATOR BIT FUNCTIONS PROPERLY.
                ;
002340  013701  TEST20: MOV     @#BASE,R1       ; GET ADDRESS
        000542
002344  012702          MOV     #<RATE1+MODE0+GO>,R2
        000011
002350  012703          MOV     #<DIO+RATE1+MODE2+GO>,R3
        004015
002354                  SCOPE                   ; DECLARE LOOP POINT
002356  005011          CLR     (R1)            ; CLEAR CSR
002360  005061          CLR     2(R1)           ; CLEAR BPR
        000002
002364  012711          MOV     #DIO,(R1)       ; DISABLE OSCILLATOR
        004000
002370  050211          BIS     R2,(R1)         ; TRY TO COUNT
002372  005000          CLR     R0              ; WAIT
002374  077001  1$:     SOB     R0,1$
002376  011100          MOV     (R1),R0         ; GET CSR
002400  010311          MOV     R3,(R1)         ; READ CLOCK
002402  052711          BIS     #MST2,(R1)      ; GENERATE ST2
        001000
002406  005761          TST     2(R1)           ; SHOULD BE ZERO
        000002
002412  001002          BNE     2$              ; OOPS, ERROR
002414  105700          TSTB    R0              ; CHECK OVERFLOW TOO
002416  100001          BPL     3$              ; GOOD - NOT SET
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 36 - CLOCK COUNTED WITH INTERNAL
                ;                       OSCILLATOR DISABLED
                ;
                ; **************************************************
                ;
002420          2$:     ERROR   36              ; REPORT ERROR
```

```
    1 002422           3$:     SCOPE                   ; DECLARE LOOP POINT
    2 002424  005011           CLR     (R1)            ; CLEAR CSR
    3 002426  005061           CLR     2(R1)           ; CLEAR BPR
              000002
    4 002432  012711           MOV     #DIO,(R1)       ; DISABLE OSCILLATOR
              004000
    5 002436  050211           BIS     R2,(R1)         ; TRY TO COUNT
    6 002440  005000           CLR     R0
    7 002442  052711     4$:   BIS     #MOSC,(R1)      ; GENERATE STROBE
              002000
    8 002446  077003           SOB     R0,4$           ; LOOP BACK
    9 002450  011100           MOV     (R1),R0         ; GET CSR
   10 002452  010311           MOV     R3,(R1)         ; READ CLOCK
   11 002454  052711           BIS     #MST2,(R1)      ; GENERATE ST2
              001000
   12 002460  005761           TST     2(R1)           ; DID IT COUNT?
              000002
   13 002464  001003           BNE     5$              ; YES - ALL DONE
   14 002466  105700           TSTB    R0              ; OVERFLOW?
   15 002470  100401           BMI     5$              ; JUST AS GOOD
   16                 ;
   17                 ; ***************************************************
   18                 ;
   19                 ;       ERROR CODE 37 - OVERFLOW BIT NOT SET
   20                 ;
   21                 ; ***************************************************
   22                 ;
   23 002472                  ERROR   37              ; REPORT ERROR
   24                 ;
   25 002474          5$:     EXIT
```

```
                        .SBTTL  TEST 21: FREQUENCY DIVIDERS
                ;
                ; THIS TEST VERIFIES THAT THE ON-BOARD FREQUENCY DIVIDER
                ; ARE FUNCTIONING PROPERLY.  THIS INSURES THE ACCURACY OF
                ; OF THE VARIOUS RATES.
                ;
002476  013701  TEST21: MOV     @#BASE,R1       ; GET ADDRESS
        000542
002502                  RELMOV  #DIVTAB,R3      ; GET TABLE ADDRESS
002510  012702          MOV     #<RATE1+MODE0+GO>,R2
        000011
002514          1$:     SCOPE                   ; DECLARE LOOP POINT
002516  011305          MOV     (R3),R5         ; GET LSW OF COUNT
002520  016304          MOV     2(R3),R4        ; GET MSW OF COUNT
        000002
002524  005011          CLR     (R1)            ; CLEAR CSR
002526  005061          CLR     2(R1)           ; CLEAR BPR
        000002
002532  052711          BIS     #DIO,(R1)       ; DISABLE OSCILLATOR
        004000
002536  050211          BIS     R2,(R1)         ; SET RATE
002540  052711  2$:     BIS     #MOSC,(R1)      ; MAINT. OSCILLATOR
        002000
002544  162705          SUB     #1,R5           ; COUNT
        000001
002550  005604          SBC     R4
002552  103372          BCC     2$
002554  011100          MOV     (R1),R0         ; SAMPLE CSR
002556                  PUSH    R2              ; SAVE RATE
002560  052716          BIS     #<DIO+MODE2+GO>,(SP)
        004005
002564                  POP     (R1)            ; LOAD CSR
002566  052711          BIS     #MST2,(R1)      ; GENERATE ST2
        001000
002572  022761          CMP     #1,2(R1)        ; CLOCK = 1?
        000001
        000002
002600  001402          BEQ     3$              ; YES - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 40 - DIVIDER DIDN'T DIVIDE
                ;
                ; ********************************************************
                ;
002602                  ERROR   40,CLOCK        ; REPORT ERROR
```

EST 21: FREQUENCY DIVIDERS

```
    1 002606             3$:     SCOPE                           ; DECLARE LOOP POINT
    2 002610  011305               MOV     (R3),R5                 ; GET LSW OF COUNT
    3 002612  016304               MOV     2(R3),R4                ; GET MSW OF COUNT
              000002
    4 002616  162705               SUB     #1,R5                   ; SUBTRACT 1
              000001
    5 002622  005604               SBC     R4
    6 002624  005011               CLR     (R1)                    ; CLEAR CSR
    7 002626  005061               CLR     2(R1)                   ; CLEAR BPR
              000002
    8 002632  052711               BIS     #DIO,(R1)               ; DISABLE OSCILLATOR
              004000
    9 002636  050211               BIS     R2,(R1)                 ; SET RATE
   10 002640  052711     4$:       BIS     #MOSC,(R1)              ; MAINT. OSCILLATOR
              002000
   11 002644  162705               SUB     #1,R5                   ; COUNT
              000001
   12 002650  005604               SBC     R4
   13 002652  103372               BCC     4$
   14 002654  011100               MOV     (R1),R0                 ; SAMPLE CSR
   15 002656                       PUSH    R2                      ; SAVE RATE
   16 002660  052716               BIS     #<DIO+MODE2+GO>,(SP)
              004005
   17 002664                       POP     (R1)                    ; LOAD CSR
   18 002666  052711               BIS     #MST2,(R1)              ; GENERATE ST2
              001000
   19 002672  005761               TST     2(R1)                   ; COUNT = 0?
              000002
   20 002676  001402               BEQ     5$                      ; YES - SKIP ERROR
   21                       ;
   22                       ; ***************************************************
   23                       ;
   24                       ;       ERROR CODE 41 - DIVIDER DIVIDED BY TOO
   25                       ;                       LITTLE (EARLY)
   26                       ;
   27                       ; ***************************************************
   28                       ;
   29 002700                       ERROR   41,CLOCK                ; REPORT ERROR
   30                       ;
   31 002704  062703     5$:       ADD     #4,R3                   ; BUMP TABLE POINTER
              000004
   32 002710  062702               ADD     #10,R2                  ; BUMP RATE
              000010
   33 002714  022702               CMP     #<RATE6+GO>,R2                  ; DONE?
              000061
   34 002720  001275               BNE     1$                      ; NO - LOOP BACK
   35 002722                       EXIT
   36                       ;
   37                       ; TABLE OF 32 BIT COUNT VALUES
   38                       ;
   39                       .NLIST  BIN
   40 002724  DIVTAB:       .WORD   11,0                    ; DECIMAL 10 - 1
   41 002730                .WORD   143,0                   ; DECIMAL 100 - 1
   42 002734                .WORD   1747,0                  ; DECIMAL 1000 - 1
   43 002740                .WORD   23417,0                 ; DECIMAL 10000 - 1
   44 002744                .WORD   103237,1                ; DECIMAL 100000 - 1
   45                               .LIST   BIN
```

```
                        .SBTTL  TEST 22: MODE 2 OPERATION
                ;
                ; THIS TEST USES THE MAINTENANCE OSCILLATOR AND THE
                ; MAINTENANCE ST2 TO CHECK MODE 2 OPERATION.
                ;
002750  013701  TEST22: MOV     @#BASE,R1       ; GET ADDRESS
        000542
002754                  SCOPE                   ; DECLARE LOOP POINT
002756  005011          CLR     (R1)            ; CLEAR CSR
002760  005061          CLR     2(R1)           ; CLEAR BAR
        000002
002764  012711          MOV     #<DIO+RATE1+MODE2+GO>,(R1)
        004015
002770  012700          MOV     #20.,R0         ; DO 20 PULSES
        000024
002774  052711  1$:     BIS     #MOSC,(R1)      ; GENERATE PULSE
        002000
003000  077003          SOB     R0,1$
003002  052711          BIS     #MST2,(R1)      ; SAVE COUNT
        001000
003006  022761          CMP     #2,2(R1)        ; COUNT = 2?
        000002
        000002
003014  001401          BEQ     2$              ; YES - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 42 - CLOCK DIDN'T COUNT
                ;
                ; ********************************************************
                ;
003016                  ERROR   42              ; REPORT ERROR
                ;
003020  042711  2$:     BIC     #ST2F,(R1)      ; CLEAR ST2 FLAG
        100000
003024  052711          BIS     #MST2,(R1)      ; GENERATE ANOTHER ST2
        001000
003030  022761          CMP     #2,2(R1)        ; COUNT STILL = 2?
        000002
        000002
003036  001401          BEQ     3$              ; YES - EXIT
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 43 - CLOCK WAS CLEARED WHEN
                ;                       IT SHOULDN'T HAVE BEEN
                ;
                ; ********************************************************
                ;
003040                  ERROR   43              ; REPORT ERROR
                ;
003042          3$:     EXIT
```

```
                        .SBTTL  TEST 23: MODE 3 OPERATION
                ;
                ; THIS TEST USES THE MAINTENANCE OSCILLATOR AND THE
                ; MAINTENANCE ST2 TO CHECK MODE 3 OPERATION
                ;
003044  013701  TEST23: MOV     @#BASE,R1       ; GET ADDRESS
        000542
003050                  SCOPE                   ; DECLARE LOOP POINT
003052  005011          CLR     (R1)            ; CLEAR CSR
003054  005061          CLR     2(R1)           ; CLEAR BPR
        000002
003060  012711          MOV     #<DIO+RATE1+MODE3+GO>,(R1)
        004017
003064  012700          MOV     #20.,R0         ; DO 20 PULSES
        000024
003070  052711  1$:     BIS     #MOSC,(R1)      ; GENERATE PULSE
        002000
003074  077003          SOB     R0,1$
003076  052711          BIS     #MST2,(R1)      ; SAVE COUNT
        001000
003102  022761          CMP     #2,2(R1)        ; COUNT = 2?
        000002
        000002
003110  001401          BEQ     2$              ; YES - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 44 - CLOCK DIDN'T COUNT
                ;
                ; ********************************************************
                ;
003112                  ERROR   44              ; REPORT ERROR
                ;
003114  042711  2$:     BIC     #ST2F,(R1)      ; CLEAR ST2 FLAG
        100000
003120  052711          BIS     #MST2,(R1)      ; GENERATE ANOTHER ST2
        001000
003124  005761          TST     2(R1)           ; COUNT = 0 NOW?
        000002
003130  001401          BEQ     3$              ; YES - EXIT
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 45 - CLOCK WASN'T CLEARED WHEN
                ;                       IT SHOULD HAVE BEEN
                ;
                ; ********************************************************
                ;
003132                  ERROR   45              ; REPORT ERROR
                ;
003134          3$:     EXIT
```

```
                        .SBTTL  TEST 24: END OF LOGIC TESTS
                ;
                ; THIS TEST IS USED TO FORCE THE TST-11 SEQUENCER TO
                ; RETURN TO COMMAND LEVEL.  THE REMAINING TESTS REQUIRE
                ; USER CONNECTIONS AND THUS MUST BE EXECUTED INDIVIDUALL
                ;
003136          TEST24: ESCAPE
                ;
                ;
                ;
                        .SBTTL  TEST 25: ST2,ST1 OUTPUTS
                ;
                ; THIS TEST PROVIDES CONTINUOUS HIGH SPEED PULSES
                ; ON THE ST2 AND ST1 OUTPUTS FOR MEASUREMENT AND
                ; OBSERVATION WITH AN OSCILLOSCOPE.
                ;
003140  013701  TEST25: MOV     @#BASE,R1       ; GET ADDRESS
        000542
003144  005011          CLR     (R1)            ; CLEAR CSR
003146                  KBEXIT                  ; SET UP KEYBOARD
003150  012711  1$:     MOV     #<MST2+MST1>,(R1)       ; STROBE
        001400
003154  000775          BR      1$              ; LOOP BACK
                ;
                ;
                ;
                        .SBTTL  TEST 26: OVERFLOW OUTPUT
                ;
                ; THIS TEST GENERATES CONTINUOUS HIGH SPEED PULSES
                ; ON THE OVERFLOW OUTPUT FOR MEASUREMENT AND
                ; OBSERVATION WITH AN OSCILLOSCOPE.  EVERY TIME
                ; A CHARACTER IS TYPED, THE RATE SELECT FIELD IS
                ; INCREMENTED BY ONE.

                ;
003156  013701  TEST26: MOV     @#BASE,R1       ; GET ADDRESS
        000542
003162  012702          MOV     #<RATE1+MODE1+GO>,R2    ; INITIAL RATE
        000013
003166  005011  1$:     CLR     (R1)            ; CLEAR CSR
003170  012761          MOV     #-1,2(R1)       ; PRESET BPR
        177777
        000002
003176  010211          MOV     R2,(R1)         ; SET UP OUTPUT
003200                  PRINT   <RATE,MODE = >
003220  010200          MOV     R2,R0
003222                  OCT8
003224                  CRLF
003226                  TTYIN                   ; WAIT FOR CHARACTER
003230  062702          ADD     #10,R2          ; NEXT RATE
        000010
003234  042702          BIC     #177700,R2      ; PREVENT CARRIES
        177700
003240  000752          BR      1$              ; LOOP BACK
```

```
                        .SBTTL  TEST 27: ST2 OUT TO ST1 IN
                ;
                ; THIS TEST REQUIRES THAT SWITCH PACK S3 BE SET
                ; ACCORDING TO THE FOLLOWING TABLE:
                ;
                ;       SWITCH  1 - OFF
                ;               2 - ON
                ;               3 - OFF
                ;               4 - OFF
                ;               5 - ON
                ;               6 - ON
                ;               7 - NOT USED
                ;               8 - NOT USED
                ;
                ; THIS SELECTS TTL THRESHOLDS AND POSITIVE SLOPE FOR
                ; SCHMITT TRIGGER 1.
                ;
                ; REMOVE ANY PREVIOUS JUMPERS AND JUMPER
                ;
                ;       PIN J1-SS  (ST2 OUT)  TO  PIN J1-VV (ST1 IN)
                ;
                ;
003242  013701  TEST27: MOV     @#BASE,R1       ; GET ADDRESS
        000542
003246                  SCOPE                   ; DECLARE LOOP POINT
003250  005011          CLR     (R1)            ; CLEAR CSR
003252  005061          CLR     2(R1)           ; CLEAR BPR
        000002
003256  012711          MOV     #<RATE6+GO>,(R1)
        000061
003262  052711          BIS     #MST2,(R1)      ; GENERATE ST2
        001000
003266  012711          MOV     #<MODE2+GO>,(R1)
        000005
003272  052711          BIS     #MST2,(R1)      ; CHECK COUNTER
        001000
003276  022761          CMP     #1,2(R1)        ; CORRECT?
        000001
        000002
003304  001401          BEQ     1$              ; YES - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 46 - ST1 WAS NOT RECEIVED,
                ;                       IT SHOULD HAVE BEEN
                ;
                ; ********************************************************
                ;
003306                  ERROR   46              ; REPORT ERROR
003310          1$:     EXIT
```

```
                        .SBTTL  TEST 30: ST1 OUT TO ST2 IN
                ;
                ; THIS TEST REQUIRES THAT SWITCH PACK S3 BE SET
                ; ACCORDING TO THE FOLLOWING TABLE:
                ;
                ;       SWITCH  1 - OFF
                ;               2 - OFF
                ;               3 - OFF
                ;               4 - ON
                ;               5 - ON
                ;               6 - ON
                ;               7 - NOT USED
                ;               8 - NOT USED
                ;
                ; THIS SELECTS TTL THRESHOLDS AND POSITIVE SLOPE FOR
                ; SCHMITT TRIGGER 2.
                ;
                ; REMOVE ANY PREVIOUS JUMPERS AND JUMPER
                ;
                ;       PIN J1-UU  (ST1 OUT)  TO   PIN J1-TT  (ST2 IN)
                ;
                ;
003312  013701  TEST30: MOV     @#BASE,R1       ; GET ADDRESS
        000542
003316  005011          CLR     (R1)            ; CLEAR CSR
003320                  SCOPE                   ; DECLARE LOOP POINT
003322  012711          MOV     #GO,(R1)        ; SET GO BIT
        000001
003326  052711          BIS     #MST1,(R1)      ; GENERATE ST1
        000400
003332  005711          TST     (R1)            ; ST2 FLAG SET?
003334  100401          BMI     1$              ; YES - SKIP ERROR
                ;
                ; ********************************************************
                ;
                ;       ERROR CODE 47 - ST2 WAS NOT RECEIVED,
                ;                       IT SHOULD HAVE BEEN
                ;
                ; ********************************************************
                ;
003336                  ERROR   47              ; REPORT ERROR
003340          1$:     EXIT
                ;
        000001          .END
```

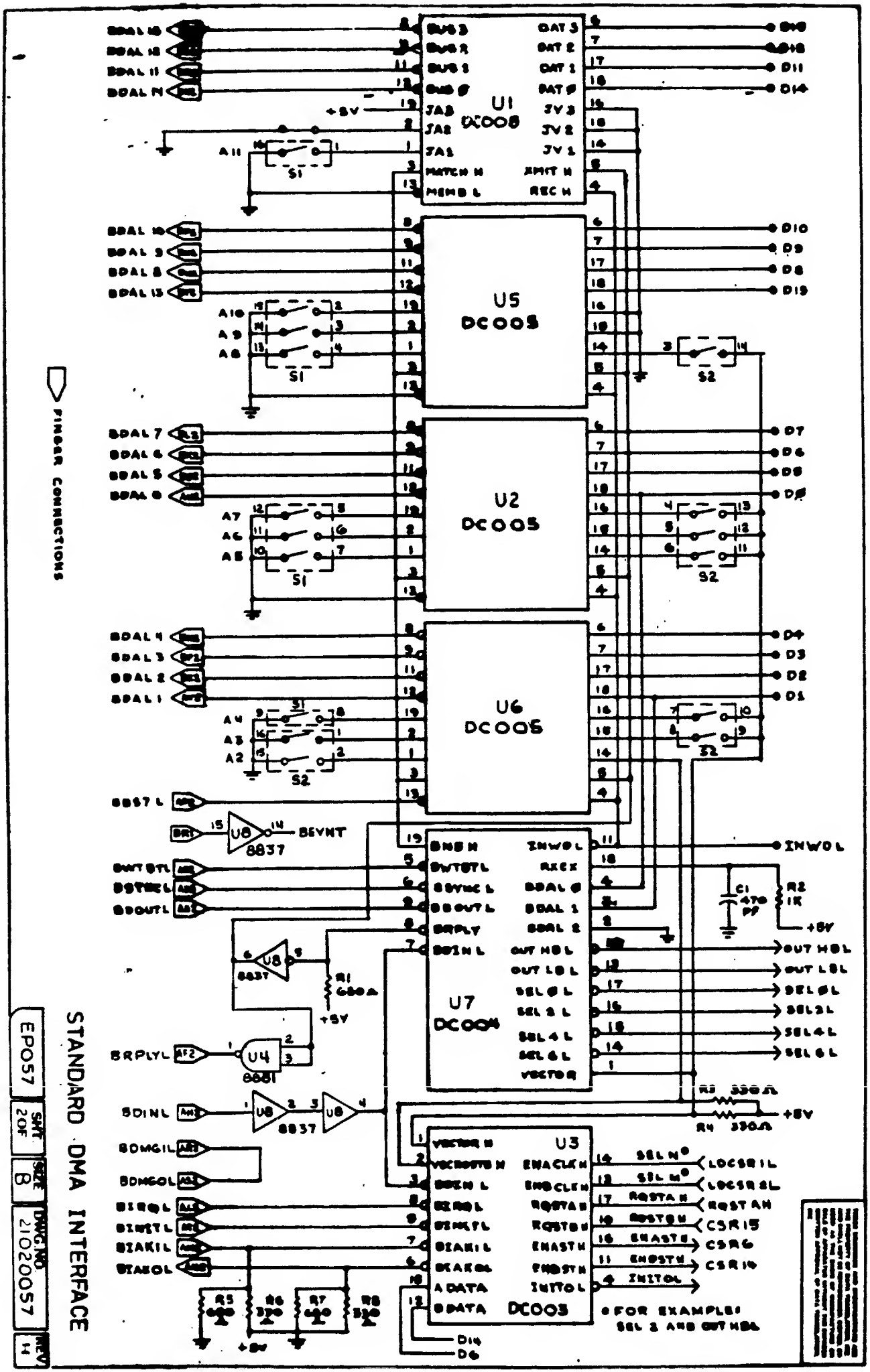
SYMBOL TABLE

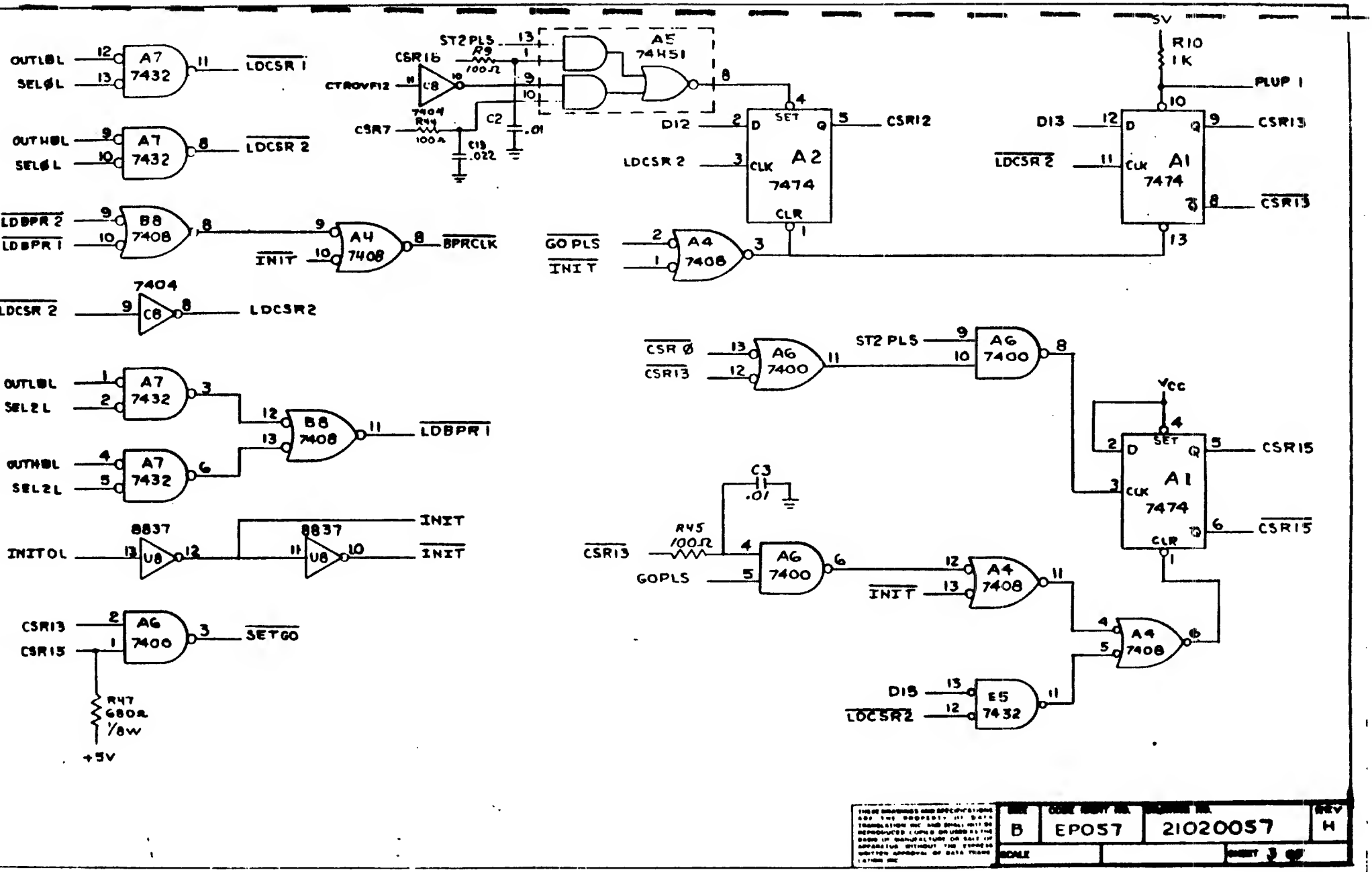
```
BASE  = 000542          I2769 = 000144RG    002 TEST10  001354R    002
BIT0  = 000001          LF    = 000012          TEST11  001462R    002
BIT1  = 000002          LPERR = 000002          TEST12  001566R    002
BIT10 = 002000          LPTST = 000001          TEST13  001650R    002
BIT11 = 004000          MODE0 = 000000          TEST14  001710R    002
BIT12 = 010000          MODE1 = 000002          TEST15  002002R    002
BIT13 = 020000          MODE2 = 000004          TEST16  002104R    002
BIT14 = 040000          MODE3 = 000006          TEST17  002204R    002
BIT15 = 100000          MOSC  = 002000          TEST2   000664R    002
BIT2  = 000004          MST1  = 000400          TEST20  002340R    002
BIT3  = 000010          MST2  = 001000          TEST21  002476R    002
BIT4  = 000020          NEWFLG= 010000          TEST22  002750R    002
BIT5  = 000040          NORPLY  000260R    002 TEST23  003044R    002
BIT6  = 000100          ODTACC= 000514          TEST24  003136R    002
BIT7  = 000200          OVFLO = 000200          TEST25  003140R    002
BIT8  = 000400          PARAM   000166R    002 TEST26  003156R    002
BIT9  = 001000          PR7   = 000340          TEST27  003242R    002
CLOCK   000474R    002  RATE0 = 000000          TEST3   000776R    002
CR    = 000015          RATE1 = 000010          TEST30  003312R    002
CSRBIT  000752R    002  RATE2 = 000020          TEST4   001070R    002
CTRLC = 000003          RATE3 = 000030          TEST5   001132R    002
CTRLI = 000011          RATE4 = 000040          TEST6   001206R    002
DIO   = 004000          RATE5 = 000050          TEST7   001266R    002
DIVTAB  002724R    002  RATE6 = 000060          TPB     000000R    002
ERRNUM= 000521          RATE7 = 000070          TSTALL= 000010
FF    = 000014          RBUF  = 177562          TSTCSR= 000500
FOR   = 010000          RCSR  = 177560          TSTNUM= 000520
GO    = 000001          REG     000334R    002 TSTONE= 000040
HLTERR= 000004          SPACE = 000040          TSTSEQ= 000020
INHERR= 000100          ST2F  = 100000          T2769 = 000000RG   002
INIT    000144R    002  ST2G  = 020000          VECTOR= 000544
INTOV = 000100          SWR   = 000540          XBUF  = 177566
INT2  = 040000          TEST1   000614R    002 XCSR  = 177564

. ABS.   000000    000
         000000    001
DT2769   003342    002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  2073 WORDS  ( 9 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  74 PAGES
DX0:2769,DX1:2769=DX1:TST11.ML/M,DX1:2769.V01
```

REVISION	ECO NO.	DATE	DWN	ENG
H	# 249-R	4-4-80	K.F.C.	K.E.D.
F-1	# 212	9-19-79	K.F.C.	K.E.D.
F	# 188	8-17-79	K.F.C.	K.E.D.
E-1	# 173	7-3-79	K.E.D.	K.E.D.
E	# 145	2-8-79	K.F.C.	K.E.D.

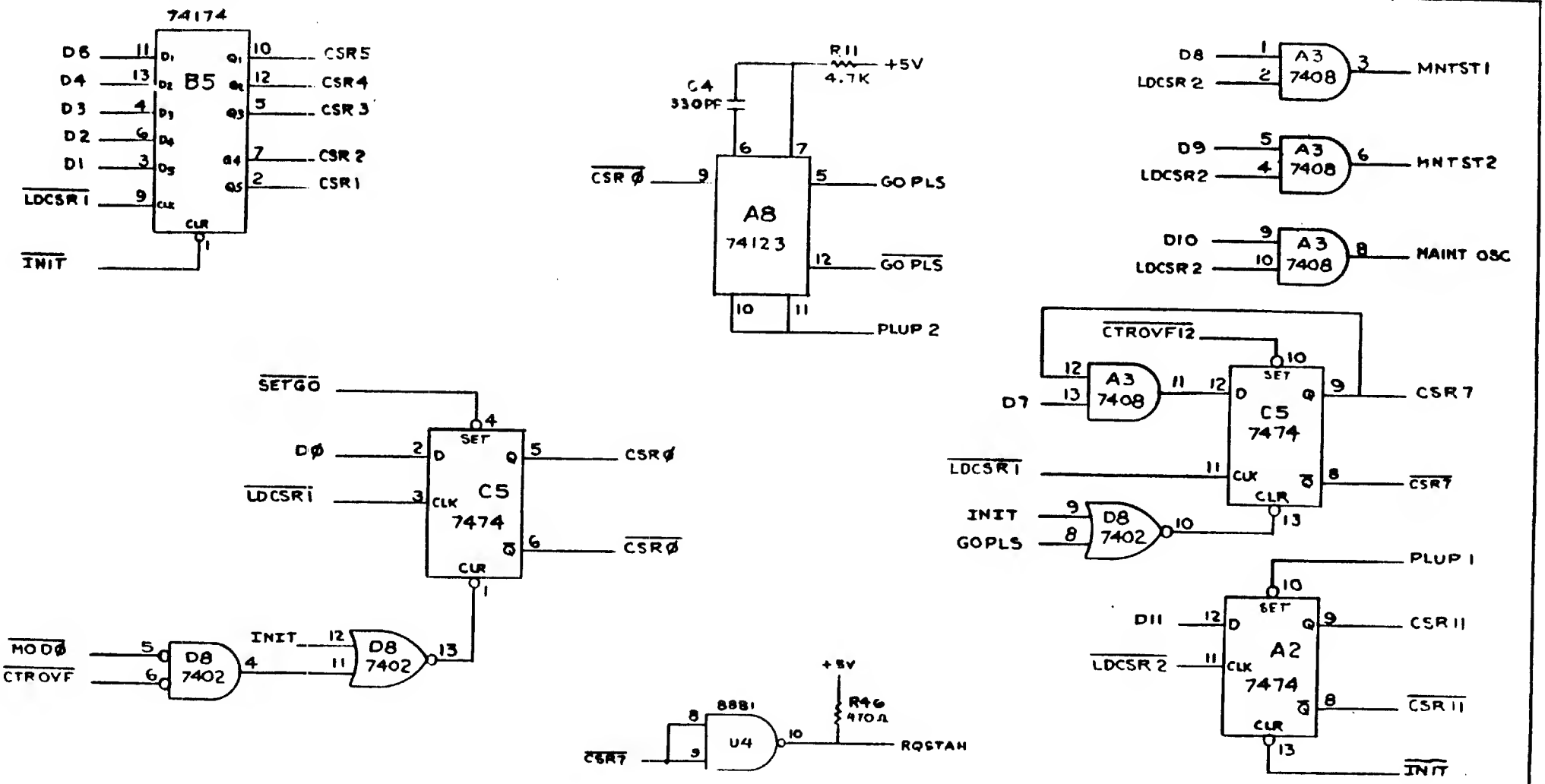
DATA TRANSLATION INC

DRAWN: K E D	DATE: 7-20-78	TITLE: SCHEMATIC REAL TIME CLOCK DT2769
CHECKED: J.W	DATE: 9-26-78	
ENG: J.W.	DATE: 9-26-78	
EP057	SHT. 1 OF 5	SIZE H / DWG.NO. 21020057 / REV. H

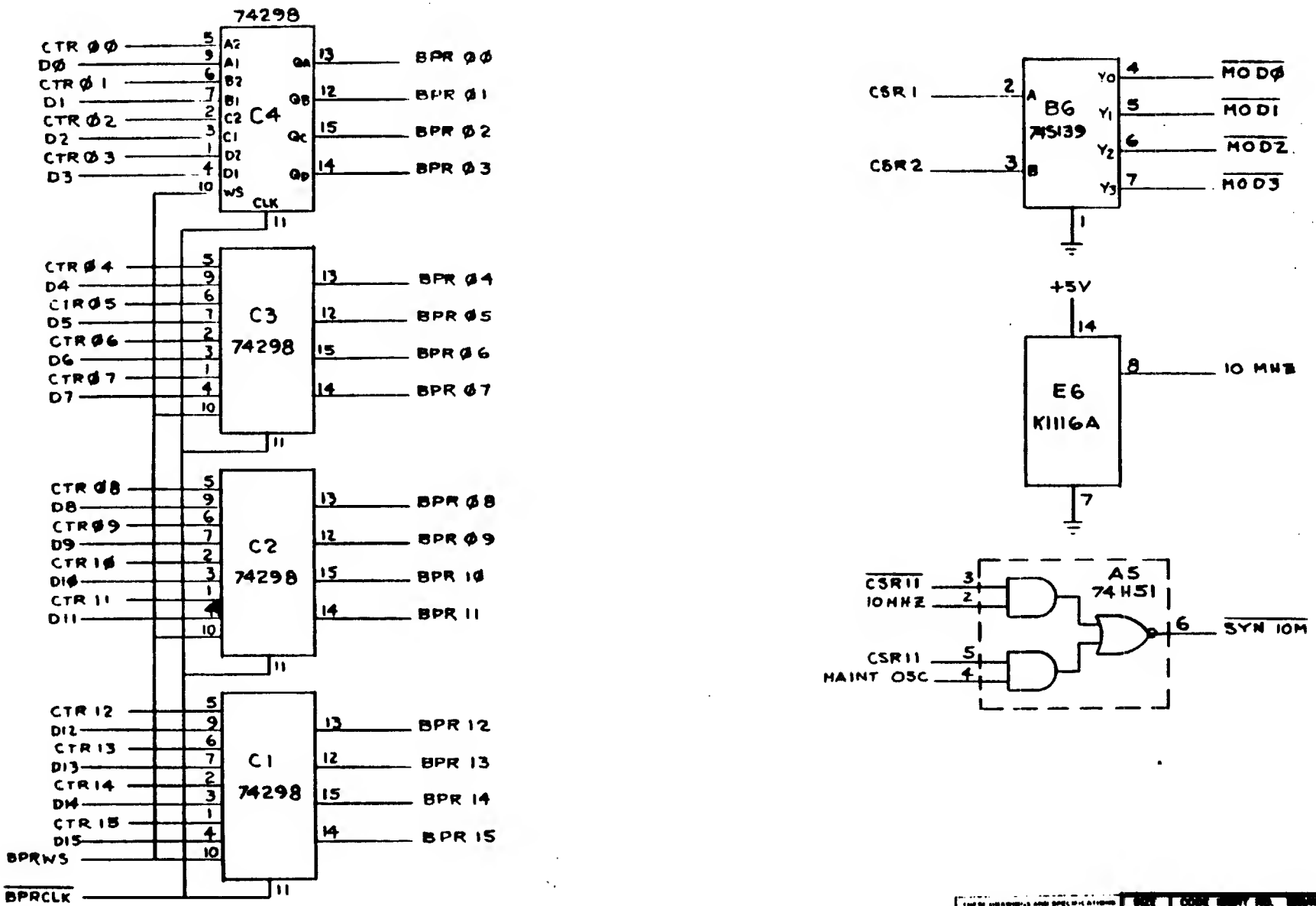

STANDARD DMA INTERFACE
FINGER CONNECTIONS
U1 DC008
U5 DC005
U2 DC005
U6 DC005
U7 DC004
U3 DC003
8837
EPO57
SHT 2OF
SIZE B
DWG NO 2102057
REV H
* FOR EXAMPLE: SEL 2 AND OUT HBL



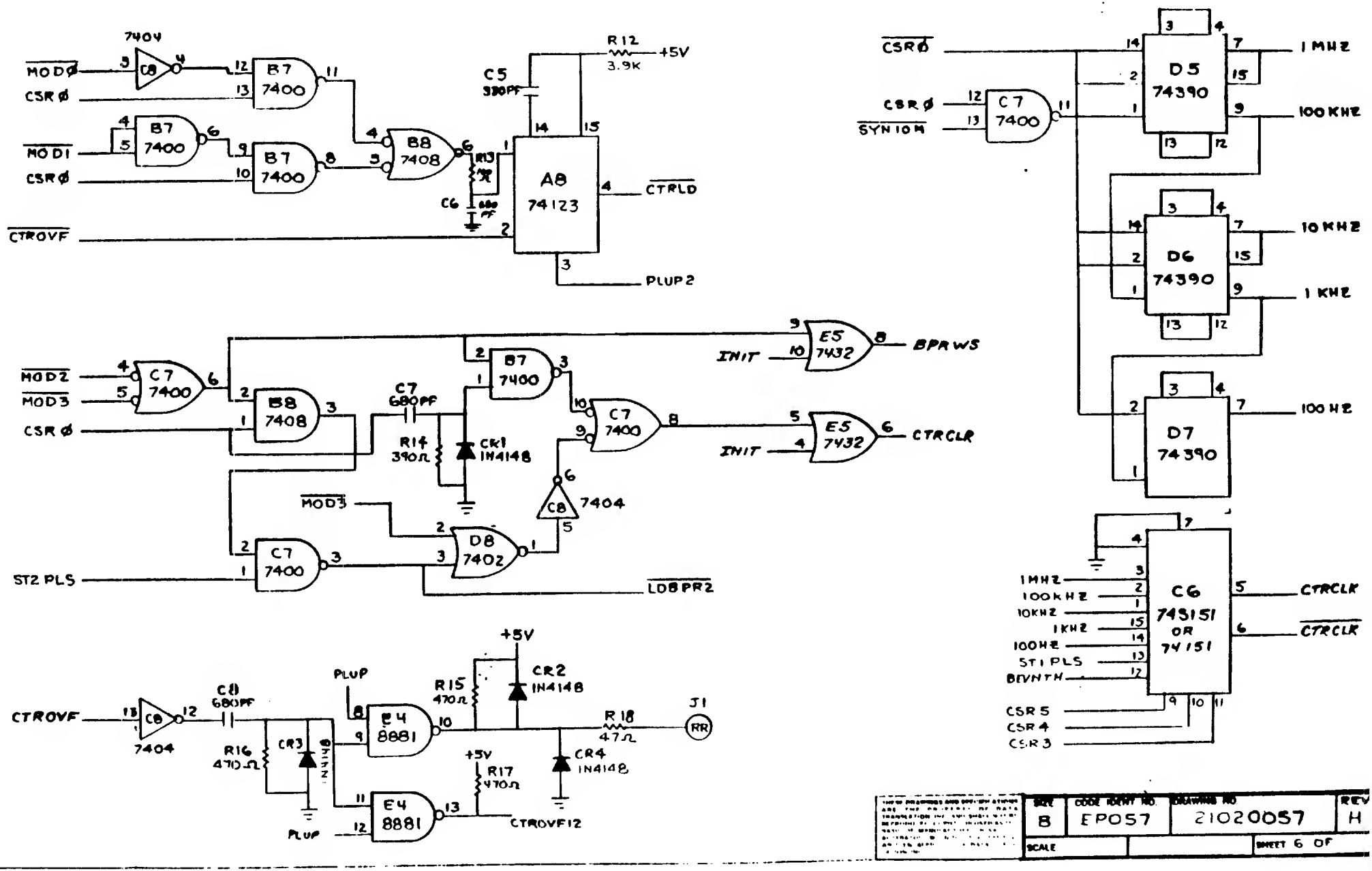
B	EP057	21020057	H



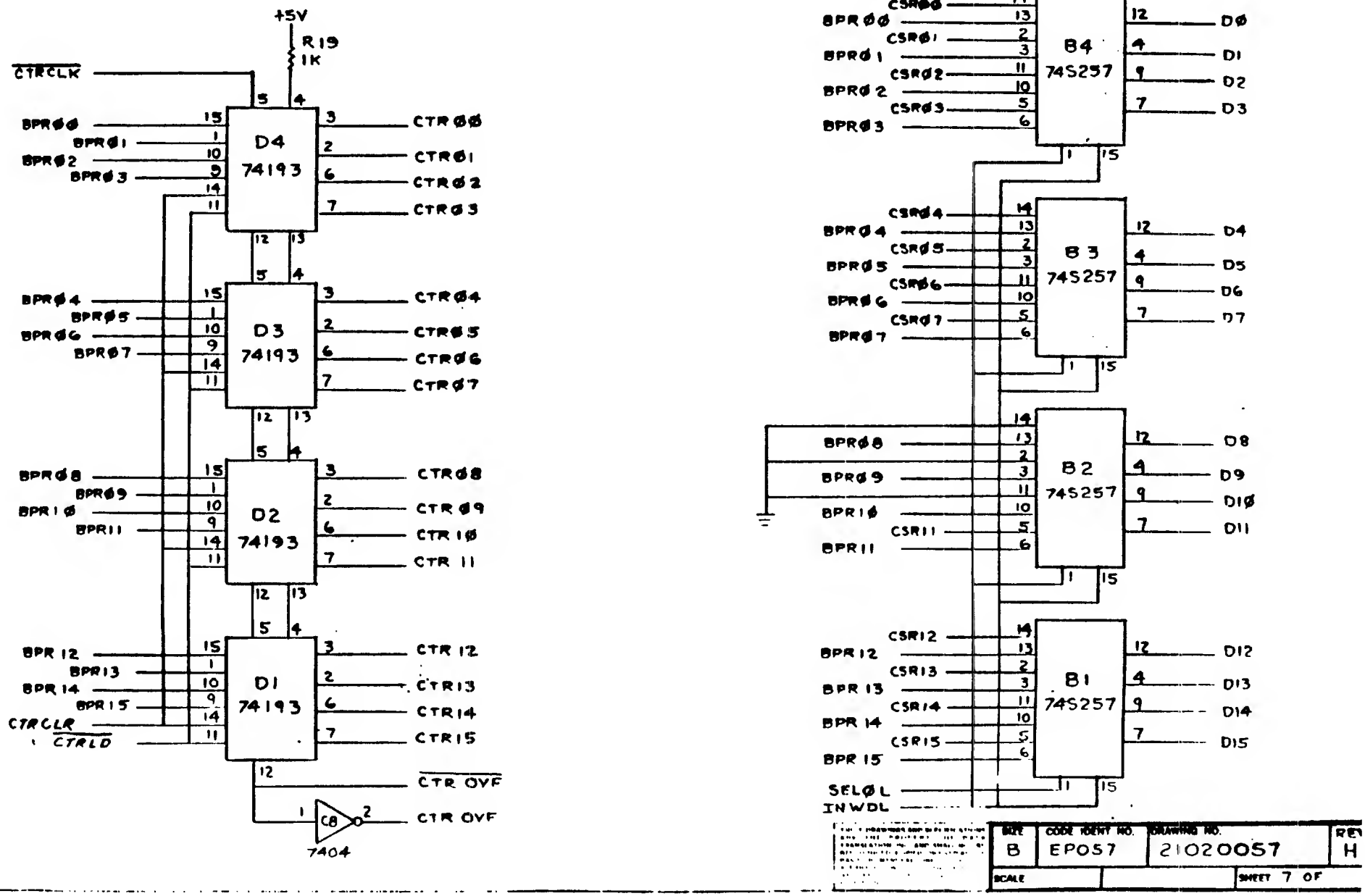
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP057	21020057	H.
SCALE			SHEET 4 OF



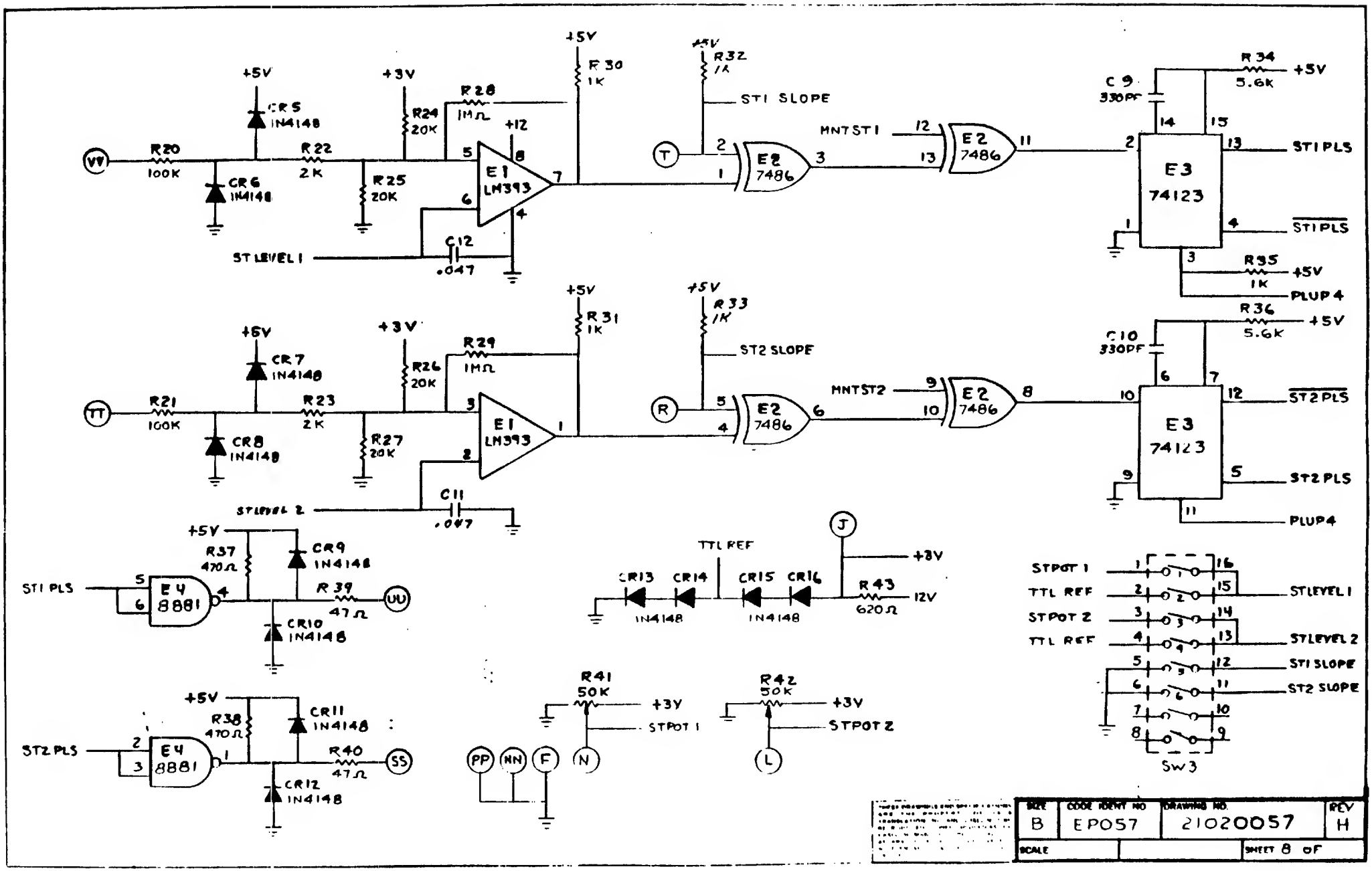
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EPOS7	21020057	H
SCALE		SHEET 5 OF	



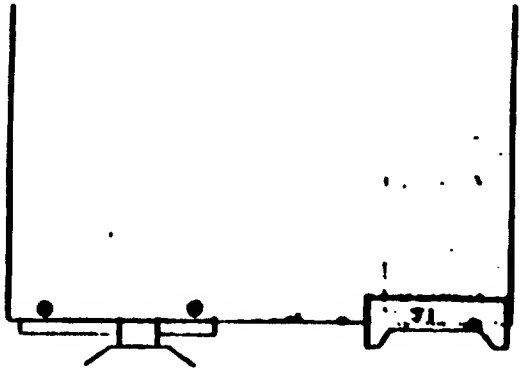
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP057	21020057	H
SCALE		SHEET 6 OF	



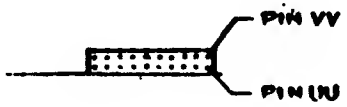
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP057	21020057	H
SCALE		SHEET 7 OF	

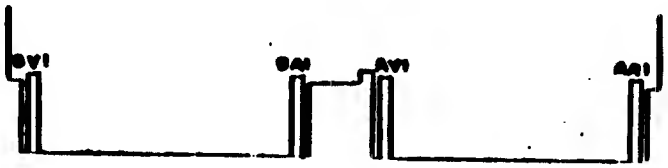


SIZE	CODE IDENT NO	DRAWING NO	REV
B	EP057	21020057	H
SCALE		SHEET 8 OF	



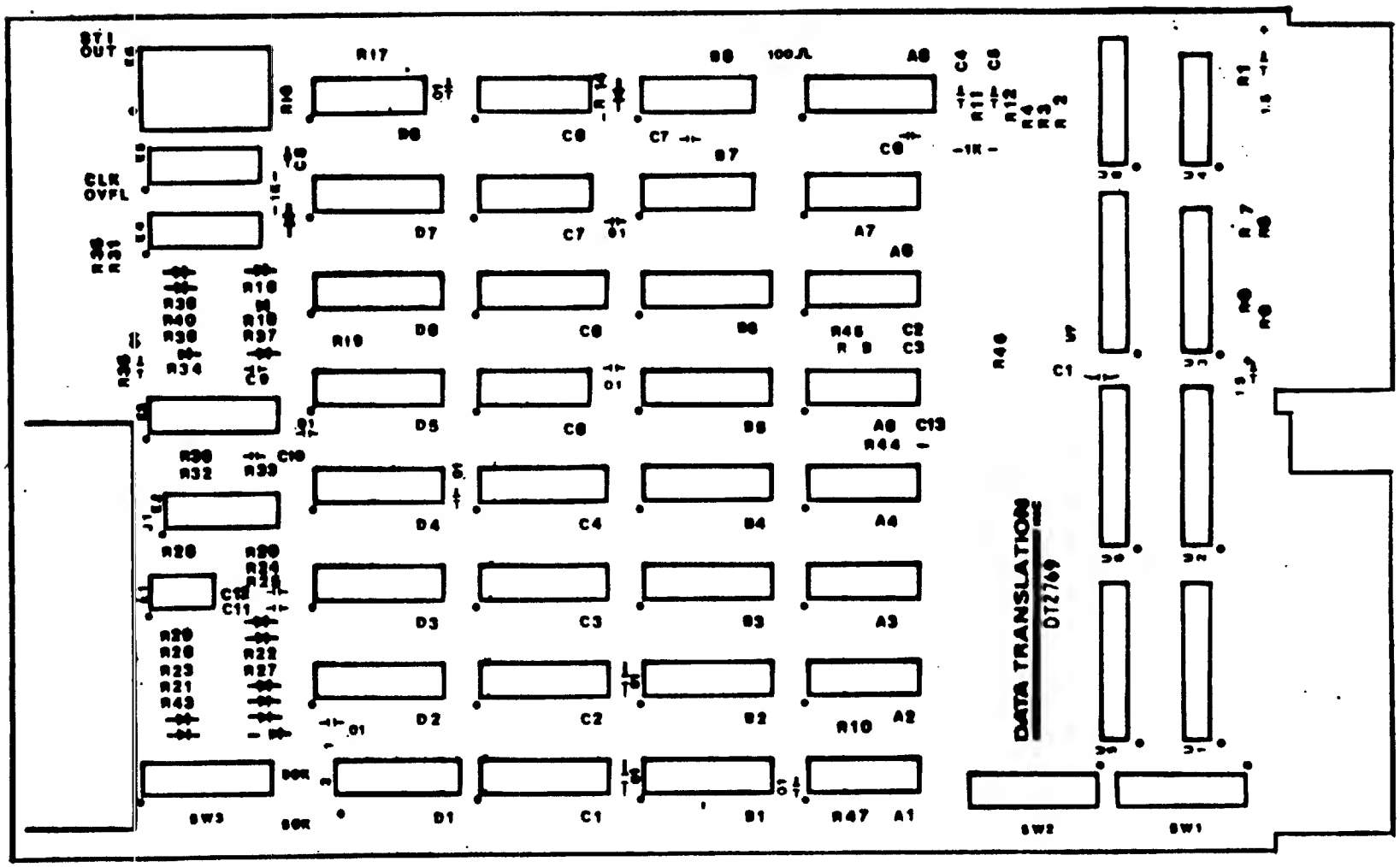
J1 CONNECTOR			
PIN	SIGNAL NAME	PIN	SIGNAL NAME
VV	ST 1 IN	VU	ST 1 OUT L
TT	ST 2 IN	SS	ST 2 OUT L
RR	CLK OVL	PP	GND
NN	GND	MM	
LL		KK	
JJ		HH	
FF		EE	
DD		CC	
BB		AA	
Z		Y	
X		W	
V		U	
T	SLOPE 1	S	
R	SLOPE 2	P	
N	POT 1	M	
L	POT 2	K	
J	+5V	H	
F	GND	E	
D		C	
B		A	





BOTTOM EDGE CONNECTOR							
COMPONENT SIDE		SOLDER SIDE		COMPONENT SIDE		SOLDER SIDE	
PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME
BA1		BA2	+5V	AA1		AA2	+5V
BB1		BB2		AB1		AB2	
BC1		BC2	0 GND	AC1		AC2	0 GND
BD1		BD2		AD1		AD2	
BE1		BE2	BDAL 3L	AE1		AE2	BDOUTL
BF1		BF2	BDAL 3L	AF1		AF2	
BH1		BH2	BDAL 4L	AH1		AH2	BDINL
BJ1	0 GND	BJ2	BDAL 5L	AJ1	0 GND	AJ2	BSYNC L
BK1		BK2	BDAL 6L	AK1		AK2	BWTBTL
BL1		BL2	BDAL 7L	AL1		AL2	BIRQL
BM1	0 GND	BM2	BDAL 8L	AM1	0 GND	AM2	BIAKIL
BN1		BN2	BDAL 9L	AN1		AN2	BIAKOL
BP1		BP2	BDAL 10L	AP1		AP2	BBS 7L
BR1		BR2	BDAL 11L	AR1		AR2	BDMGIL
BS1		BS2	BDAL 12L	AS1		AS2	BDMGOL
BT1	0 GND	BT2	BDAL 13L	AT1	0 GND	AT2	BINITL
BU1		BU2	BDAL 14L	AU1		AU2	
BV1	+5V	BV2	BDAL 15L	AV1		AV2	

	SIZE	CODE IDENT NO	DRAWING NO	REV
	B	EP057	21020057	H
	SCALE			SHT 9 OF 9



COMPONENT DESIGNATIONS

SIZE	CODE IDENT NO.	DRAWING NO.	REV
A	EP057	11070057	H
SCALE			SHEET 1 OF 1

DATA TRANSLATION
INC
4 Strathmore Rd., Natick MA 01760
(617) 655-5300 Telex 948474